

## 40G/100G MULTISERVICE TEST MODULE

# FTB/IQS-85100G

## Packet Blazer



---

Copyright © 2012–2017 EXFO Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form, be it electronically, mechanically, or by any other means such as photocopying, recording or otherwise, without the prior written permission of EXFO Inc. (EXFO).

Information provided by EXFO is believed to be accurate and reliable. However, no responsibility is assumed by EXFO for its use nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent rights of EXFO.

EXFO's Commerce And Government Entities (CAGE) code under the North Atlantic Treaty Organization (NATO) is 0L8C3.

The information contained in this publication is subject to change without notice.

***Trademarks***

EXFO's trademarks have been identified as such. However, the presence or absence of such identification does not affect the legal status of any trademark.

***Units of Measurement***

Units of measurement in this publication conform to SI standards and practices.

February 13, 2017

Software version: 2.16

Document version: 9.0.0.0

---

# Contents

<b>1</b>	<b>Introducing the FTB/IQS-85100G SCPI Commands</b> .....	<b>1</b>
	Conventions .....	2
<b>2</b>	<b>Getting Started</b> .....	<b>3</b>
<b>3</b>	<b>Communicating Through TCP/IP over Telnet</b> .....	<b>7</b>
	Introducing TCP/IP over Telnet .....	7
	Features .....	8
	Configuring Your Unit and Modules to Work With TCP/IP over Telnet .....	8
	Executing SCPI Commands Over Telnet .....	14
	Releasing Modules .....	20
	Internal Commands of the TCP/IP over Telnet Protocol .....	21
<b>4</b>	<b>SCPI Command List - General</b> .....	<b>29</b>
	Delay Between SCPI Commands .....	29
	Test Applications .....	29
	Test Information and Control .....	29
	Discover Remote Button .....	30
	About (i) Button .....	31

---

<b>5 SCPI Command List - Setup .....</b>	<b>33</b>
List of Pages .....	33
BERT and Unframed BERT .....	35
CFP/CFP2 .....	38
Clock .....	39
EtherBERT, and Unframed BERT .....	41
EtherSAM - Burst .....	45
EtherSAM - Global .....	46
EtherSAM - Ramp .....	48
FTFL/PT .....	49
Frequency .....	51
GFP-F/GFP-T .....	52
Interface (Ethernet) .....	53
Labels .....	54
MAC/IP/UDP .....	55
Modify Structure (Transport) .....	60
Modify Structure (Ethernet) .....	62
Network .....	63
ODU Channels - Global .....	65
RFC 2544 - Global .....	67
RFC 2544 - Subtests .....	69
RFC 6349 .....	72
Services - Global .....	74
Services - Profile .....	76
Signal (Transport) .....	79
Signal - Signal Configuration (OTN) .....	81
Signal - Signal Configuration (SONET/SDH) .....	82
Smart Loopback .....	83
Streams - Global .....	84
Streams - Profile .....	87
System .....	89
Test Configurator .....	90
Timer .....	91
Traces (OTN) .....	92
Traces (SONET/SDH) .....	96

---

<b>6 SCPI Command List - Results</b> .....	<b>99</b>
List of Pages .....	99
Alarms/Errors .....	100
FTFL/PT .....	134
GFP-F/GFP-T .....	136
Labels .....	137
Logger and Alarms/Errors Logger .....	138
MPLS .....	139
OTL-SDT .....	140
Performance Monitoring .....	141
SDT (Multi-Channel OTN) .....	142
Service Configuration - Burst .....	144
Service Configuration - Ramp .....	145
Service Performance .....	146
Streams - Frame Loss / Out-of-Sequence .....	148
Streams - Jitter .....	149
Streams - Latency .....	150
Streams - Throughput .....	151
Summary .....	152
Summary (EtherSAM) .....	155
Summary (Multi-Channel OTN) .....	157
Summary (RFC 2544) .....	158
Summary (RFC 6349) .....	160
Summary (Traffic Gen & Mon) .....	162
Traces - OTN .....	163
Traces - SONET/SDH .....	167
Traffic - Ethernet .....	169
Traffic - Flow Control .....	170

---

<b>7</b>	<b>SCPI Command List - Functions</b>	<b>171</b>
	List of Pages	171
	40/100G Advanced - CFP/CFP2/CFP4/QSFP Control	172
	40/100G Advanced - Lanes Mapping & Skew	174
	40/100G Advanced - Pre-Emphasis	175
	APS	177
	Client Offset	179
	Filters	180
	GMP	185
	OH - GFP-F/GFP-T	186
	OH - OTN	188
	OH - SONET/SDH	189
	Packet Capture	191
	Ping & Trace Route	193
	Pointer Adjustment	196
	RTD	199
	Traffic Scan	200
<b>8</b>	<b>SCPI Command List - Pop-Up</b>	<b>201</b>
	List of Pages	201
	Bulk Read	202
	Config TCM	203
	Configure Per Frame Size	204
	Copy Service	205
	Copy Stream	206
	Filter Configuration	207
	IPv6 Address Configuration	210
	Laser ON/OFF Button	212
	Manual Mapping	213
	Manual Skew	214
	Modify Frame Structure	215
	Modify Tributary Slots/Port	217
	Modify Trib Slots/Channels (Multi-Channel OTN)	218
	Profile (Services)	219
	Profile (Stream)	220
	Shaping	221
	Stream (Summary)	222
	Thresholds (RFC 2544)	223
	TOS/DS Configuration	224
	Triggered Frame Details	225

<b>9 SCPI Command Reference</b> .....	<b>227</b>
LINStrument <LogicalInstrumentPos> .....	227
Test Control .....	229
:FETCh[1..n]:DATA:TELEcom:TEST:TIME? .....	229
:FETCh[1..n]:DATA:TELEcom:MODUle:DETAils:CDATe? .....	230
:FETCh[1..n]:DATA:TELEcom:MODUle:DETAils:SPVersion? .....	231
:SOURce[1..n]:DATA:TELEcom:VERDict:ENABle .....	232
:SOURce[1..n]:DATA:TELEcom:VERDict:ENABle? .....	233
:SOURce[1..n]:DATA:TELEcom:REStore:DEFault .....	234
:SOURce[1..n]:DATA:TELEcom:TEST .....	235
:SOURce[1..n]:DATA:TELEcom:TEST? .....	236
:SOURce[1..n]:DATA:TELEcom:RESet .....	237
:CONFig[1..n]:DATA:TELEcom:LOAD .....	238
:CONFig[1..n]:DATA:TELEcom:SAVE .....	239
:CONFig[1..n]:WAIT:TIME .....	240
:FETCh[1..n]:DATA:TELEcom:OPTical:MODUle:STATus? .....	241
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:STATus .....	242
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:STATus? .....	243
:SOURce[1..n]:DATA:TELEcom:TEST:TYPE .....	244
:SOURce[1..n]:DATA:TELEcom:TEST:TYPE? .....	245
Modify Structure .....	246
:SOURce[1..n]:DATA:TELEcom:ITYPE .....	246
:SOURce[1..n]:DATA:TELEcom:ITYPE? .....	248
:SOURce[1..n]:DATA:TELEcom:ETHernet:BERT:FRAMing .....	249
:SOURce[1..n]:DATA:TELEcom:ETHernet:BERT:FRAMing? .....	250
:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver .....	251
:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver? .....	252
:SOURce[1..n]:DATA:TELEcom:ODU:TYPE .....	253
:SOURce[1..n]:DATA:TELEcom:ODU:TYPE? .....	257
:SOURce[1..n]:DATA:TELEcom:OTN:FRAMing .....	262
:SOURce[1..n]:DATA:TELEcom:OTN:FRAMing? .....	263
:SOURce[1..n]:DATA:TELEcom:OTN:CLlEnt .....	264
:SOURce[1..n]:DATA:TELEcom:OTN:CLlEnt? .....	265
:SOURce[1..n]:DATA:TELEcom:TOPology .....	266
:SOURce[1..n]:DATA:TELEcom:TOPology? .....	267
:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:TRANsparent:MODE:ENABle .....	268
:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:TRANsparent:MODE:ENABle? .....	269
:SOURce[1..n]:DATA:TELEcom:SONet:TEST:TYPE .....	270
:SOURce[1..n]:DATA:TELEcom:SONet:TEST:TYPE? .....	271
:FETCh[1..n]:DATA:TELEcom:SDHSONet:FRAMing? .....	272
:FETCh[1..n]:DATA:TELEcom:SDHSONet:CLlEnt? .....	273
:SOURce[1..n]:DATA:TELEcom:HOP:TYPE .....	274
:SOURce[1..n]:DATA:TELEcom:HOP:TYPE? .....	275
:SOURce[1..n]:DATA:TELEcom:OTN:MULTiplex:ITYPE .....	276
:SOURce[1..n]:DATA:TELEcom:OTN:MULTiplex:ITYPE? .....	277
Clock .....	278
:INPut[1..n]:TELEcom:BACKplane:CLOCK .....	278

:INPut[1..n]:TELEcom:BACKplane:CLOCK?	279
:INPut[1..n]:TELEcom:LBO	280
:INPut[1..n]:TELEcom:LBO?	281
:OUTPut[1..n]:TELEcom:TERMination	282
:OUTPut[1..n]:TELEcom:TERMination?	283
:OUTPut[1..n]:TELEcom:LEVel	284
:OUTPut[1..n]:TELEcom:LEVel?	285
:INPut[1..n]:TELEcom:LEVel	286
:INPut[1..n]:TELEcom:LEVel?	287
:OUTPut[1..n]:TELEcom:FRAMing	288
:OUTPut[1..n]:TELEcom:FRAMing?	289
:INPut[1..n]:TELEcom:FRAMing	290
:INPut[1..n]:TELEcom:FRAMing?	291
:OUTPut[1..n]:TELEcom:CODE	292
:OUTPut[1..n]:TELEcom:CODE?	293
:INPut[1..n]:TELEcom:CODE	294
:INPut[1..n]:TELEcom:CODE?	295
:OUTPut[1..n]:TELEcom:CLOCK:FREQuency?	296
:OUTPut[1..n]:TELEcom:CLOCK:FREQuency:OFFSet?	297
:INPut[1..n]:TELEcom:COUtput:FREQuency?	298
:INPut[1..n]:TELEcom:COUtput:STATUs?	299
:INPut[1..n]:TELEcom:BCLock:ENABle	300
:INPut[1..n]:TELEcom:BCLock:ENABle?	301
:INPut[1..n]:TELEcom:BACKplane:STATUs?	302
:INPut[1..n]:TELEcom:CLOCK:ALARm:STATUs?	303
:OUTPut[1..n]:TELEcom:CLOCK:ALARm:STATUs?	304
<b>CFP/CFP2</b>	<b>305</b>
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:MODUle:ID?	305
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:VENdor:NAME?	306
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:PART:NUMBer?	307
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:SERial:NUMBer?	308
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:REVision?	309
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:FIRMware:VERsion?	310
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:CONNector:TYPE?	311
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:SPEEd?	312
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:TYPE?	313
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:POWer:CLASs?	314
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:LRAtio:TYPE?	315
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:WDM:TYPE?	316
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:CLEI:PRESENce?	317
:SENSE[1..n]:DATA:TELEcom:OPTical:CFP:MODE?	318
<b>Signal (Transport)</b>	<b>319</b>
:SENSE[1..n]:DATA:TELEcom:OPTical:TX:POWer?	319
:SENSE[1..n]:DATA:TELEcom:OPTical:RX:POWer?	320
:SENSE[1..n]:DATA:TELEcom:OPTical:RX:POWer:MINimum?	321
:SENSE[1..n]:DATA:TELEcom:OPTical:RX:POWer:MAXimum?	322
:SENSE[1..n]:DATA:TELEcom:OPTical:WAVelength?	323
:OUTPut[1..n]:TELEcom:LASer	324



:OUTPut[1..n]:TELEcom:LASer? .....	325
:SENSe[1..n]:DATA:TELEcom:OPTical:POWEr:RANGe? .....	326
:SENSe[1..n]:DATA:TELEcom:LOFF .....	327
:SENSe[1..n]:DATA:TELEcom:LOFF? .....	328
:FETCh[1..n]:DATA:TELEcom:LINK:GLOBal:STATus? .....	329
:SOURce[1..n]:DATA:TELEcom:OTN:FEC .....	330
:SOURce[1..n]:DATA:TELEcom:OTN:FEC? .....	331
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler .....	332
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler? .....	333
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler .....	334
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler? .....	335
:SOURce[1..n]:DATA:TELEcom:OTN:BTRaffic:PT[1..n] .....	336
:SOURce[1..n]:DATA:TELEcom:OTN:BTRaffic:PT[1..n]? .....	337
<b>Frequency .....</b>	<b>338</b>
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency? .....	338
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue .....	339
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue? .....	340
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet .....	341
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet? .....	342
:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency? .....	343
:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:POSitive? .....	344
:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:NEGAtive? .....	345
:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue? .....	346
<b>Traces (OTN) .....</b>	<b>347</b>
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16 .....	347
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16? .....	348
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16 .....	349
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16? .....	350
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B32 .....	351
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B32? .....	352
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B .....	353
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B? .....	354
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B .....	355
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B? .....	356
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:OPSPec:B .....	357
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:OPSPec:B? .....	358
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:EXPEcted .....	359
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:EXPEcted? .....	360
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:EXPEcted .....	361
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:EXPEcted? .....	362
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPEcted .....	363
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPEcted? .....	364
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPEcted .....	365
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPEcted? .....	366
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM .....	367
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM? .....	368
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM .....	369
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM? .....	370

:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:SAPI:B .....	371
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:SAPI:B? .....	372
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:DAPI:B .....	373
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:DAPI:B? .....	374
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:OPSPec:B .....	375
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:OPSPec:B? .....	376
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:SAPI:B .....	377
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:SAPI:B? .....	378
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:DAPI:B .....	379
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:DAPI:B? .....	380
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:OPSPec:B .....	381
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:OPSPec:B? .....	382
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPEcted .....	383
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPEcted? .....	384
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXPEcted .....	385
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXPEcted? .....	386
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:EXPEcted .....	387
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:EXPEcted? .....	388
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:EXPEcted .....	389
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:EXPEcted? .....	390
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM .....	391
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM? .....	392
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:TIM .....	393
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:TIM? .....	394
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:GOVErwrite? .....	395
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:GOVErwrite? .....	396
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHannel .....	397
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHannel? .....	398
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHANnel .....	399
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHANnel? .....	400
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B .....	401
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B? .....	402
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B .....	403
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B? .....	404
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:OPSPec:B .....	405
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:OPSPec:B? .....	406
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:SAPI:B .....	407
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:SAPI:B? .....	408
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:DAPI:B .....	409
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:DAPI:B? .....	410
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:OPSPec:B .....	411
:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:OPSPec:B? .....	412
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted .....	413
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted? .....	414
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPEcted .....	415
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPEcted? .....	416
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted .....	417
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted? .....	418

:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted	419
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted?	420
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:TIM	421
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:TIM?	422
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:TIM	423
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:TIM?	424
<b>FTFL/PT</b>	<b>425</b>
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDIcAtion	425
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDIcAtion?	427
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE	428
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE?	429
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier	430
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier?	431
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPeC	432
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPeC?	433
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDIcAtion	434
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDIcAtion?	435
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE	436
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE?	437
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENtifier	438
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENtifier?	439
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:OPSPeC	440
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:OPSPeC?	441
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe	442
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?	445
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE	448
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?	449
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe	450
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?	453
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE	456
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?	457
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PLM	459
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PLM?	460
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe	461
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe?	464
:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe:RECeived?	467
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:GOVErwrite?	468
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHANnel	469
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHANnel?	470
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHAnnel	471
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHAnnel?	472
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe	473
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe?	476
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE	479
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE?	480
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE	481
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE?	482
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PLM	483

:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PLM?	484
:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE:RECEived?	485
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:COPY	486
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE	487
<b>MAC/IP/UDP</b>	<b>488</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRESS:DESTination	488
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRESS:DESTination?	489
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRESS:DESTination:IP	490
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRESS:DESTination:IP?	491
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRESS:SOURCE?	492
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRESS:SOURCE:IP	493
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRESS:SOURCE:IP?	494
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve	495
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve?	496
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve:STAUs?	497
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:QPING	498
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:QPING?	499
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ETHER?	500
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:IP:AUTomatic:STATus	501
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:IP:AUTomatic:STATus?	502
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway	503
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway?	504
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDRESS	505
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDRESS?	506
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL	507
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL?	508
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:MASK:IP	509
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:MASK:IP?	510
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID	511
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID?	513
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE	515
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE?	517
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:PRIORITY	519
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:PRIORITY?	520
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID:ELIGiblebit	521
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID:ELIGiblebit?	522
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT	523
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT?	524
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:PORT	526
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:PORT?	527
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs	529
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs?	530
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:IP:MULTiplicat	531
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:IP:MULTiplicat?	532
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:IP:RANGE	533
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:IP:RANGE?	534
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PAYLoad	535
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PAYLoad?	536

:SOURCE[1..n]:DATA:TELECOM:ETHERNET:ESAM:CONFIG:SERVICES:VLAN:TYPE .....	537
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:ESAM:CONFIG:SERVICES:VLAN:TYPE? .....	540
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:ESAM:CONFIG:SERVICES:VLAN:ID .....	542
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:ESAM:CONFIG:SERVICES:VLAN:ID? .....	544
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:ESAM:CONFIG:SERVICES:VLAN:ID:ELIGIBLEBIT .....	547
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:ESAM:CONFIG:SERVICES:VLAN:ID:ELIGIBLEBIT? .....	549
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:ESAM:CONFIG:SERVICES:VLAN:PRIORITY .....	551
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:ESAM:CONFIG:SERVICES:VLAN:PRIORITY? .....	553
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:ADDRESS:DESTINATION:FLOODING .....	556
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:ADDRESS:DESTINATION:FLOODING? .....	557
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:ADDRESS:FLOODING:RANGE .....	558
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:ADDRESS:FLOODING:RANGE? .....	559
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:ADDRESS:SOURCE:FLOODING .....	560
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:ADDRESS:SOURCE:FLOODING? .....	561
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:SOURCE:IPVERSION:MULTIPLICAT .....	562
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:SOURCE:IPVERSION:MULTIPLICAT? .....	563
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:SOURCE:IPVERSION:RANGE .....	564
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:SOURCE:IPVERSION:RANGE? .....	565
:SENSE[1..n]:DATA:TELECOM:ETHERNET:STREAM:DESTINATION:IPVERSION .....	566
:SENSE[1..n]:DATA:TELECOM:ETHERNET:STREAM:DESTINATION:IPVERSION? .....	567
:SENSE[1..n]:DATA:TELECOM:ETHERNET:STREAM:FLABEL:IPVERSION .....	568
:SENSE[1..n]:DATA:TELECOM:ETHERNET:STREAM:FLABEL:IPVERSION? .....	569
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:IPVERSION:HOP:LIMIT .....	570
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:IPVERSION:HOP:LIMIT? .....	571
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:MAC:OUI .....	573
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:MAC:OUI? .....	574
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:MPLS:LABEL .....	575
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:MPLS:LABEL? .....	576
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:MPLS:COSEXP .....	578
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:MPLS:COSEXP? .....	580
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:MPLS:TTL .....	582
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:MPLS:TTL? .....	583
EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT .....	585
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:RATE .....	585
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:RATE? .....	586
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:TX:STATUS .....	587
:SOURCE[1..n]:DATA:TELECOM:ETHERNET:STREAM:TX:STATUS? .....	588
:SOURCE[1..n]:DATA:TELECOM:PATTERN:TYPE .....	589
:SOURCE[1..n]:DATA:TELECOM:PATTERN:TYPE? .....	590
:SENSE[1..n]:DATA:TELECOM:PATTERN:TYPE .....	591
:SENSE[1..n]:DATA:TELECOM:PATTERN:TYPE? .....	592
:SOURCE[1..n]:DATA:TELECOM:POLARITY .....	593
:SOURCE[1..n]:DATA:TELECOM:POLARITY? .....	594
:SENSE[1..n]:DATA:TELECOM:POLARITY .....	595
:SENSE[1..n]:DATA:TELECOM:POLARITY? .....	596
:SENSE[1..n]:DATA:TELECOM:COUPLED .....	597
:SENSE[1..n]:DATA:TELECOM:COUPLED? .....	598
:SENSE[1..n]:DATA:TELECOM:PATTERN:RXANALYSIS:STATUS .....	599

:SENSe[1..n]:DATA:TELEcom:PATtern:RXPanalysis:STATUS? .....	600
:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue .....	601
:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue? .....	602
:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue .....	603
:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue? .....	604
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL .....	605
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL? .....	606
:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:TX .....	607
:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:TX? .....	608
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX .....	609
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX? .....	610
:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:TX .....	611
:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:TX? .....	612
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:RX .....	613
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:RX? .....	614
:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX .....	615
:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX? .....	617
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:RX .....	619
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:RX? .....	621
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:COUPled .....	623
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:COUPled? .....	624
:FETCh[1..n]:DATA:TELEcom:EOTN:ALARm:LINK? .....	625
:SENSe[1..n]:DATA:TELEcom:SDT:NTTime .....	626
:SENSe[1..n]:DATA:TELEcom:SDT:NTTime? .....	627
<b>RFC 2544 - Global .....</b>	<b>628</b>
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution .....	628
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution? .....	629
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity .....	630
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity? .....	631
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZE .....	632
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZE? .....	633
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:RESTore .....	634
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MINTime? .....	635
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABle .....	636
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABle? .....	637
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MINTime? .....	638
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ENABle .....	639
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ENABle? .....	640
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:MINTime? .....	641
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:ENABle .....	642
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:ENABle? .....	643
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MINTime? .....	644
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ENABle .....	645
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ENABle? .....	646
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TOTal:MINTime? .....	647
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:VALidation? .....	648
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIRection .....	649
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIRection? .....	650

:SOURce[1..n]:DATA:TELEcom:ETHernet:DUALtest:ENABLEd .....	651
:SOURce[1..n]:DATA:TELEcom:ETHernet:DUALtest:ENABLEd? .....	652
<b>Smart Loopback .....</b>	<b>653</b>
:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MODE .....	653
:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MODE? .....	654
:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MATChing:UDP:PORT:MODE? .....	655
:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MATChing:MAC:ADDRes:MODE? .....	656
<b>BERT .....</b>	<b>657</b>
:FETCh[1..n]:DATA:TELEcom:PATTern:ALARm:SYNC? .....	657
:FETCh[1..n]:DATA:TELEcom:PATTern:GLOBal:ALARm:SYNC? .....	658
:SENSe[1..n]:DATA:TELEcom:PATTern:THReshold:RATE .....	659
:SENSe[1..n]:DATA:TELEcom:PATTern:THReshold:RATE? .....	660
:SENSe[1..n]:DATA:TELEcom:PATTern:THReshold:COUNt .....	661
:SENSe[1..n]:DATA:TELEcom:PATTern:THReshold:COUNt? .....	662
:SOURce[1..n]:DATA:TELEcom:PATTern:VERDict:DISable .....	663
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATTern:THReshold:RATE .....	664
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATTern:THReshold:RATE? .....	665
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATTern:THReshold:COUNt .....	666
:SENSe[1..n]:DATA:TELEcom:UPRBs:PATTern:THReshold:COUNt? .....	667
:SOURce[1..n]:DATA:TELEcom:OTN:REStore:DEFault .....	668
:SENSe[1..n]:DATA:TELEcom:SDT:NDTime .....	669
:SENSe[1..n]:DATA:TELEcom:SDT:NDTime? .....	670
:SENSe[1..n]:DATA:TELEcom:SDT .....	671
:SENSe[1..n]:DATA:TELEcom:SDT? .....	672
:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE .....	673
:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE? .....	674
:SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection .....	675
:SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection? .....	676
:SENSe[1..n]:DATA:TELEcom:SDT:VERDict .....	677
:SENSe[1..n]:DATA:TELEcom:SDT:VERDict? .....	678
:SENSe[1..n]:DATA:TELEcom:SDT:THReshold .....	679
:SENSe[1..n]:DATA:TELEcom:SDT:THReshold? .....	680
<b>Interface - Laser ON/OFF .....</b>	<b>681</b>
:SENSe[1..n]:DATA:TELEcom:ALASer .....	681
:SENSe[1..n]:DATA:TELEcom:ALASer? .....	682
:SENSe[1..n]:DATA:TELEcom:LASer .....	683
:SENSe[1..n]:DATA:TELEcom:LASer? .....	684
<b>Signal - Signal Configuration (OTN) - Modify Tributary Slots/Port .....</b>	<b>685</b>
:SOURce[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABLE .....	685
:SOURce[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABLE? .....	686
:SOURce[1..n]:DATA:TELEcom:OTN:PORT .....	687
:SOURce[1..n]:DATA:TELEcom:OTN:PORT? .....	688
:SOURce[1..n]:DATA:TELEcom:OTN:POSition .....	689
:SOURce[1..n]:DATA:TELEcom:OTN:POSition? .....	690
:SOURce[1..n]:DATA:TELEcom:OTN:POSition:RANGe .....	691
:SOURce[1..n]:DATA:TELEcom:OTN:POSition:RANGe? .....	692
:FETCh[1..n]:DATA:TELEcom:OTN:BITRate? .....	693

:FETCh[1..n]:DATA:TELEcom:OTN:SLOTs? .....	694
Signal - Signal Configuration (OTN) - Config TCM .....	695
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:CONFig:TCM[1..n] .....	695
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:CONFig:TCM[1..n]? .....	696
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:CONFig:TCM[1..n] .....	697
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:CONFig:TCM[1..n]? .....	698
Modify Frame Structure .....	699
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DATalink .....	699
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DATalink? .....	700
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork .....	701
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork? .....	702
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANSport? .....	703
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN .....	704
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN? .....	705
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACked .....	706
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACked? .....	707
:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN .....	708
:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN? .....	709
:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STACked .....	711
:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STACked? .....	712
:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion .....	713
:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion? .....	714
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS .....	715
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS? .....	716
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:HEADers .....	717
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:HEADers? .....	718
TOS/DS Configuration .....	719
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS .....	719
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS? .....	720
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE .....	721
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE? .....	722
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:ECN .....	725
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:ECN? .....	726
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:BIT .....	727
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:BIT? .....	728
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:COST .....	729
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:COST? .....	730
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DELay .....	731
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DELay? .....	732
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECEdence .....	733
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECEdence? .....	735
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:RELIability .....	736
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:RELIability? .....	737
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:THROUGHput .....	738
:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:THROUGHput? .....	739
Configure Per Frame Size .....	740
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ALL:FRAME .....	740



:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LAteNcy:ALL:FRAME? .....	741
GFP-F/GFP-T .....	742
:SOURCE[1..n]:DATA:TELEcom:GFP:CONFig:EXI .....	742
:SOURCE[1..n]:DATA:TELEcom:GFP:CONFig:EXI? .....	743
:SOURCE[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:TYPE .....	744
:SOURCE[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:TYPE? .....	745
:SOURCE[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:CID .....	746
:SOURCE[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:CID? .....	747
Signal - Signal Configuration (SONET/SDH) .....	748
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:S:BITS:SSMessage .....	748
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:S:BITS:SSMessage? .....	750
:SOURCE[1..n]:DATA:TELEcom:BACKground:COMPutation .....	752
:SOURCE[1..n]:DATA:TELEcom:BACKground:COMPutation? .....	753
:SOURCE[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP .....	754
:SOURCE[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP? .....	755
:SOURCE[1..n]:DATA:TELEcom:BACKground:BULK .....	756
:SOURCE[1..n]:DATA:TELEcom:BACKground:BULK? .....	757
:SOURCE[1..n]:DATA:TELEcom:POSITION .....	758
:SOURCE[1..n]:DATA:TELEcom:POSITION? .....	760
:SENSe[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TCUNeq:ENABLE .....	761
:SENSe[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TCUNeq:ENABLE? .....	762
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABLE .....	763
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABLE? .....	764
:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABLE .....	765
:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABLE? .....	766
Traces (SONET/SDH) .....	767
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern .....	767
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern? .....	768
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern:B .....	769
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern:B? .....	770
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATTern .....	771
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATTern? .....	772
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATTern:B .....	773
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATTern:B? .....	774
:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM .....	775
:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM? .....	776
:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATTern .....	777
:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATTern? .....	778
:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATTern:B .....	779
:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATTern:B? .....	780
:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM .....	781
:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM? .....	782
:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATTern .....	783
:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATTern? .....	784
:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATTern:B .....	785
:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATTern:B? .....	786
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:MESSAge .....	787
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:MESSAge? .....	788

:SENSE[1..n]:DATA:TELEcom:SDHSONet:HOP:TCAPident:EXPEcted	789
:SENSE[1..n]:DATA:TELEcom:SDHSONet:HOP:TCAPident:EXPEcted?	790
:SENSE[1..n]:DATA:TELEcom:SDHSONet:HOP:TCAPident:TCTim	791
:SENSE[1..n]:DATA:TELEcom:SDHSONet:HOP:TCAPident:TCTim?	792
<b>Streams - Profile</b>	<b>793</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:MODE	793
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:MODE?	794
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame	795
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame?	796
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE:TYPE	797
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE:TYPE?	798
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE	799
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE?	800
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MODE	801
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MODE?	802
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:COUNT	803
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:COUNT?	804
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:STEP	805
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:STEP?	806
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:TIME	807
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:TIME?	808
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled	809
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled?	810
:SOURCE[1..n]:DATA:TELEcom:ETHernet:TOTal:BANDwidth?	811
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ENABled:BANDwidth?	812
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FCOUNT	813
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FCOUNT?	814
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:RATE	815
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:RATE?	816
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:START	817
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:START?	818
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:END	819
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:END?	820
<b>Streams - Profile (Profile)</b>	<b>821</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE	821
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE?	822
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VOICe	823
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VOICe?	824
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VOICe:CALLs	825
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VOICe:CALLs?	826
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VIDeo	827
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VIDeo?	828
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VIDeo:CHANnELs	829
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VIDeo:CHANnELs?	830
<b>Shaping</b>	<b>831</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:BANDwidth	831
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:BANDwidth?	832
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNT	833

:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNT?	834
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:TIME	835
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:TIME?	836
<b>Streams - Global</b>	<b>837</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:REStore:DEfault	837
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABle	838
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABle?	839
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NAME	840
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NAME?	841
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABle	842
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABle?	843
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDReSS:TYPE	844
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDReSS:TYPE?	845
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP	846
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP?	847
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:SUBNet:MASk	848
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:SUBNet:MASk?	849
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEfault:GATeway:ENABle	850
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEfault:GATeway:ENABle?	851
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEfault:GATeway:IP	852
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEfault:GATeway:IP?	853
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEStination:IP:ENABle	854
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEStination:IP:ENABle?	855
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEStination:IP	856
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEStination:IP?	857
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEStination:MAC:ENABle	858
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEStination:MAC:ENABle?	859
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEStination:MAC:TYPE	860
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEStination:MAC:TYPE?	861
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEStination:MAC:ADDReSS	862
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEStination:MAC:ADDReSS?	863
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy	864
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:STReam	865
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:STReam?	866
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:APPLy	867
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:SYNC:PROGReSS?	868
<b>Streams - Global (Copy Stream)</b>	<b>869</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:COpyStream	869
<b>Services - Profile</b>	<b>870</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:TRAPolicing:ENABle	870
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:TRAPolicing:ENABle?	871
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:TRAPolicing:VALue	872
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:TRAPolicing:VALue?	874
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:INFRate:ENABle	876
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:INFRate:ENABle?	877
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:INFRate	879
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:INFRate?	881
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:PERCriteriA:ENABle	884

:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:ENABle? .....	886
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VALue .....	888
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VALue? .....	891
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe:ENABle .....	894
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe:ENABle? .....	895
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe .....	897
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe? .....	899
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:BMRate .....	902
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:BMRate? .....	903
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle .....	905
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle? .....	906
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:RESDefault .....	907
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMESize .....	908
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMESize? .....	909
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity .....	911
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity? .....	912
<b>EtherSAM - Global .....</b>	<b>913</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE .....	913
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE? .....	914
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBAL:SPRTTest:DURation .....	915
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBAL:SPRTTest:DURation? .....	916
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBAL:SPRTTest:ENABled .....	917
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBAL:SPRTTest:ENABled? .....	918
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBAL:TDURation:ESTimate? .....	919
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBAL:PDIRection:CONFig:STATus .....	920
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBAL:PDIRection:CONFig:STATus? .....	921
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RESTore:DEFault .....	922
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBAL:LMMode .....	923
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBAL:LMMode? .....	924
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBAL:LATency:ALARm:CURRent? .....	925
<b>EtherSAM - Burst .....</b>	<b>926</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSequence .....	926
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSequence? .....	927
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERatio .....	928
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERatio? .....	929
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame .....	930
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame? .....	931
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:CBs:TIME? .....	932
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:EBS:TIME? .....	933
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:TBURSt:TIME? .....	934
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:TOTAL? .....	935
<b>EtherSAM - Ramp .....</b>	<b>936</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:ADD .....	936
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DELete .....	937
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DEFault .....	938
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME .....	939
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME? .....	940

:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:DURation? .....	941
<b>Labels</b> .....	<b>942</b>
:SOURce[1..n]:DATA:TELEcom:SDHSONet:HOP:PATH:LABel .....	942
:SOURce[1..n]:DATA:TELEcom:SDHSONet:HOP:PATH:LABel? .....	945
:SENSE[1..n]:DATA:TELEcom:SDHSONet:HOP:PATH:LABel:EXPEcted .....	948
:SENSE[1..n]:DATA:TELEcom:SDHSONet:HOP:PATH:LABel:EXPEcted? .....	951
:SENSE[1..n]:DATA:TELEcom:SDHSONet:HOP:PUNeq .....	953
:SENSE[1..n]:DATA:TELEcom:SDHSONet:HOP:PUNeq? .....	954
<b>Thresholds (RFC 2544)</b> .....	<b>955</b>
:SOURce[1..n]:DATA:TELEcom:ETHernet:REMOte:THREshold .....	955
:SOURce[1..n]:DATA:TELEcom:ETHernet:REMOte:THREshold? .....	956
<b>Network</b> .....	<b>958</b>
:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:ADDREss:IP .....	958
:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:ADDREss:IP? .....	959
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DHCP:STATus .....	960
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DHCP:STATus? .....	961
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK .....	962
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK? .....	963
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:STATus .....	964
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:STATus? .....	965
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:ADDREss .....	966
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:ADDREss? .....	967
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:MAC:ADDREss .....	968
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:MAC:ADDREss? .....	969
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN .....	970
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN? .....	971
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:STACKed .....	972
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:STACKed? .....	973
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID .....	974
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID? .....	975
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:TYPE .....	976
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:TYPE? .....	977
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:PRIority .....	978
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:PRIority? .....	979
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID:ELIGiblebit .....	980
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID:ELIGiblebit? .....	981
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault .....	982
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault? .....	983
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DATALink:TYPE? .....	984
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAL:IPVersion:ADDREss .....	985
:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAL:IPVersion:ADDREss? .....	986
<b>RFC 2544 - Subtests</b> .....	<b>987</b>
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THROughput:MAXRate .....	987
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THROughput:MAXRate? .....	988
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THROughput:ACCURacy .....	990
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THROughput:ACCURacy? .....	991
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THROughput:AERRors .....	992
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THROughput:AERRors? .....	993

:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:TAVerage .....	994
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:TAVerage? .....	995
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:VALidations .....	996
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:VALidations? .....	997
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:TTIME .....	998
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:TTIME? .....	999
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames .....	1000
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames? .....	1001
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy .....	1002
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy? .....	1003
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors .....	1004
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors? .....	1005
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage .....	1006
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage? .....	1007
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:NBURst? .....	1008
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TTIME .....	1009
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TTIME? .....	1010
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TGRanularity .....	1011
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TGRanularity? .....	1012
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TAVerage .....	1013
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TAVerage? .....	1014
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:MAXRate .....	1015
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:MAXRate? .....	1016
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TAVerage .....	1018
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TAVerage? .....	1019
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate[1..n] .....	1020
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate[1..n]? .....	1021
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COpytest .....	1023
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COpytest? .....	1024
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate:SETall .....	1025
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate:SETall? .....	1026
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MARGin .....	1028
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MARGin? .....	1029
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TTIME .....	1030
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TTIME? .....	1031
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:LMMode .....	1032
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:LMMode? .....	1033
IPv6 Address Configuration .....	1034
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCal:IPVersion:MODE .....	1034
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCal:IPVersion:MODE? .....	1035
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:ADDRes .....	1036
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:ADDRes? .....	1037
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:MODE .....	1038
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:MODE? .....	1039
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:PMASK .....	1040
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:PMASK? .....	1041
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:IICoupled .....	1042
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:IICoupled? .....	1043

:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:ADDRESS .....	1044
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:ADDRESS? .....	1045
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:MODE .....	1046
:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:MODE? .....	1047
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAl:IPVersion:ADDRESS .....	1048
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAl:IPVersion:ADDRESS? .....	1049
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAl:IPVersion:MODE .....	1050
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAl:IPVersion:MODE? .....	1051
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LOCAl:IPVersion:ADDRESS:STATus? .....	1052
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAl:IPVersion:ADDRESS .....	1053
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAl:IPVersion:ADDRESS? .....	1054
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAl:IPVersion:MODE .....	1055
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAl:IPVersion:MODE? .....	1056
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAl:IPVersion:PMASK .....	1057
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAl:IPVersion:PMASK? .....	1058
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAl:IPVersion:IICoupled .....	1059
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAl:IPVersion:IICoupled? .....	1060
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBAl:IPVersion:ADDRESS:STATus? .....	1061
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:ADDRESS .....	1062
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:ADDRESS? .....	1063
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:MODE .....	1064
:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:MODE? .....	1065
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:DGATeway:IPVersion:ADDRESS:STATus? .....	1066
<b>RFC 6349 .....</b>	<b>1067</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE .....	1067
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE? .....	1068
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:DIRection .....	1069
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:DIRection? .....	1070
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort .....	1071
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort? .....	1072
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:WSIZetarget .....	1073
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:WSIZetarget? .....	1074
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:CIR .....	1075
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:CIR? .....	1076
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TOSDs .....	1077
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TOSDs? .....	1078
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:MULTiple:CONNections .....	1079
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:MULTiple:CONNections? .....	1080
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:MAX:MTU .....	1081
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:MAX:MTU? .....	1082
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:PATH:MTU:DISCcovery .....	1083
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:PATH:MTU:DISCcovery? .....	1084
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:DURation .....	1085
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:DURation? .....	1086
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold .....	1087
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold? .....	1088
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:VERDict .....	1089
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:VERDict? .....	1090

:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep .....	1091
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep? .....	1092
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:DURation .....	1093
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:DURation? .....	1094
:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:RESTore:DEFault .....	1095
<b>Modify Trib Slots/Channels (Multi-Channel OTN) .....</b>	<b>1096</b>
:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries .....	1096
:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries? .....	1099
:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:DEFault .....	1101
:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COpy .....	1102
:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled .....	1103
:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled? .....	1104
<b>Timer .....</b>	<b>1105</b>
:SOURce[1..n]:DATA:TELEcom:TImer:CONFig .....	1105
:SOURce[1..n]:DATA:TELEcom:TImer:CONFig? .....	1106
:SOURce[1..n]:DATA:TELEcom:TImer:STARt .....	1107
:SOURce[1..n]:DATA:TELEcom:TImer:STARt? .....	1108
:SOURce[1..n]:DATA:TELEcom:TImer:STOP .....	1109
:SOURce[1..n]:DATA:TELEcom:TImer:STOP? .....	1110
:SOURce[1..n]:DATA:TELEcom:TImer .....	1111
:SOURce[1..n]:DATA:TELEcom:TImer? .....	1112
:SOURce[1..n]:DATA:TELEcom:TImer:DURation .....	1113
:SOURce[1..n]:DATA:TELEcom:TImer:DURation? .....	1114
:SOURce[1..n]:DATA:TELEcom:TImer:UDEF .....	1115
:SOURce[1..n]:DATA:TELEcom:TImer:UDEF? .....	1116
<b>System .....</b>	<b>1117</b>
:SOURce[1..n]:DATA:TELEcom:FACTory:RESTore:DEFault .....	1117
<b>Summary .....</b>	<b>1118</b>
:FETCh[1..n]:DATA:TELEcom:TEST:STARt:TIME? .....	1118
:FETCh[1..n]:DATA:TELEcom:TEST:STATus? .....	1119
:FETCh[1..n]:DATA:TELEcom:TEST:POWer:RECOvery:COUnT? .....	1120
:FETCh[1..n]:DATA:TELEcom:TEST:STATus:VERDict? .....	1121
:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern .....	1122
:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern? .....	1123
:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:SEConds? .....	1124
:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:HISTory? .....	1125
:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:CURrent? .....	1126
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:SEConds? .....	1127
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:COUnT? .....	1128
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:RATE? .....	1129
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:HISTory? .....	1130
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:CURrent? .....	1131
:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE? .....	1132
:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm .....	1133
:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm? .....	1134
:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:MANual:TYPE? .....	1135
:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:MANual:AMOUNT .....	1136
:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:MANual:AMOUNT? .....	1137



:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:MANual:INJect .....	1138
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:TYPE? .....	1139
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:RATE .....	1140
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:RATE? .....	1141
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:CONTInuous .....	1142
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:CONTInuous? .....	1143
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated .....	1144
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated? .....	1145
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:LANE .....	1146
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:LANE? .....	1147
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALANes .....	1148
:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALANes? .....	1149
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:THReshold:VERDICT? .....	1150
:FETCh[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:VERDICT? .....	1151
:FETCh[1..n]:DATA:TELEcom:SDT:SHORtest? .....	1152
:FETCh[1..n]:DATA:TELEcom:SDT:LONGest? .....	1153
:FETCh[1..n]:DATA:TELEcom:SDT:LAST? .....	1154
:FETCh[1..n]:DATA:TELEcom:SDT:AVERAge? .....	1155
:FETCh[1..n]:DATA:TELEcom:SDT:TOTAL? .....	1156
:FETCh[1..n]:DATA:TELEcom:SDT:COUNt? .....	1157
:FETCh[1..n]:DATA:TELEcom:SDT:DEFect? .....	1158
:FETCh[1..n]:DATA:TELEcom:SDT:VERDICT? .....	1159
:FETCh[1..n]:DATA:TELEcom:OTL:SUMMary:SDT:LONGest? .....	1160
:FETCh[1..n]:DATA:TELEcom:OTL:SUMMary:SDT:SHORtest? .....	1161
:FETCh[1..n]:DATA:TELEcom:OTL:SUMMary:SDT:LAST? .....	1162
:FETCh[1..n]:DATA:TELEcom:OTL:SUMMary:SDT:AVERAge? .....	1163
:FETCh[1..n]:DATA:TELEcom:OTL:SUMMary:SDT:TOTAL? .....	1164
:FETCh[1..n]:DATA:TELEcom:OTL:SUMMary:SDT:COUNt? .....	1165
<b>Summary (Traffic Gen &amp; Mon) - (Stream) .....</b>	<b>1166</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:COUNt:FRAMes:RX? .....	1166
:FETCh[1..n]:DATA:TELEcom:ETHernet:COUNt:FRAMes:TX? .....	1167
<b>Summary (EtherSAM) .....</b>	<b>1168</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:STATus? .....	1168
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:VERDICT? .....	1169
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:VERDICT? .....	1170
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SPRTest:VERDICT? .....	1171
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOsSs? .....	1172
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter? .....	1173
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency? .....	1174
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXRate? .....	1175
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOsSs:VERDICT? .....	1176
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter:VERDICT? .....	1177
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency:VERDICT? .....	1178
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXRate:VERDICT? .....	1179
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTest:FLOsSs? .....	1180
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTest:MAXJitter? .....	1181
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTest:MLATency? .....	1182
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTest:ARXRRate? .....	1183

:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTest:FLOSS:VERDict?	1184
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTest:MAXJitter:VERDict?	1185
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTest:MLATency:VERDict?	1186
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTest:ARXRate:VERDict?	1187
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:VLAN:PREServ?	1188
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SERVices:VLAN:PREServ?	1189
<b>Summary (RFC 2544)</b>	<b>1190</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TSTate?	1190
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:SMESsage?	1191
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:FCOunt:TX?	1192
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:FCOunt:RX?	1193
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TRESults[1..n]?	1194
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:CTRial?	1197
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TSTate?	1198
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:SMESsage?	1199
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:FCOunt:TX?	1200
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:FCOunt:RX?	1201
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:CTRial?	1202
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:BBRresults[1..n]?	1203
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TSTate?	1206
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:SMESsage?	1207
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:FCOunt:TX?	1208
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:FCOunt:RX?	1209
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:CTRial?	1210
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:CSTep?	1211
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:FRESult[1..n]?	1212
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TSTate?	1214
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:SMESsage?	1215
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:CTRial?	1216
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:LRESults[1..n]?	1217
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:FCOunt:TX?	1220
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:FCOunt:RX?	1221
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TETime?	1222
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TETime?	1223
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TETime?	1224
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TETime?	1225
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:SUMMary:THReshold:VERDict?	1226
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:THReshold:VERDict?	1228
<b>Summary (Traffic Gen &amp; Mon)</b>	<b>1229</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:STATus:TIME?	1229
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:HISTory?	1230
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:CURRent?	1231
<b>Summary (RFC 6349)</b>	<b>1232</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:WINDow?	1232
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:ACTUal:L?	1233
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:IDEal:L?	1234
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:TCP:EFFiciency?	1235
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:BUFFer:DELay?	1236

:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:L:ACTUal? .....	1237
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:ACTUal:L:VERDicT? .....	1238
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold? .....	1239
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:MTU? .....	1240
:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:MINimum:RTT? .....	1241
<b>Alarms/Errors .....</b>	<b>1242</b>
:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:SECOnds? .....	1242
:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:HISTory? .....	1243
:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CURREnt? .....	1244
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:SECOnds? .....	1245
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:COUNt? .....	1246
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:RATE? .....	1247
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AMOUnt .....	1248
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AMOUnt? .....	1249
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE .....	1250
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE? .....	1251
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:INJect .....	1252
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CHAN .....	1253
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CHAN? .....	1254
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:HISTory? .....	1255
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CURREnt? .....	1256
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE .....	1257
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE? .....	1258
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:RATE .....	1259
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:RATE? .....	1260
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:CONTInuous .....	1261
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:CONTInuous? .....	1262
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated .....	1263
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated? .....	1264
:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:TYPE? .....	1265
:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CHAN .....	1266
:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CHAN? .....	1267
:INPut[1..n]:TELEcom:BACKplane:ALARm:STATus:HISTory? .....	1268
:INPut[1..n]:TELEcom:BACKplane:ALARm:STATus:CURREnt? .....	1269
:INPut[1..n]:TELEcom:BACKplane:ALARm:STATus:SECOnds? .....	1270
:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:HISTory? .....	1271
:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:CURREnt? .....	1272
:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:HISTory? .....	1273
:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:CURREnt? .....	1274
:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:RX:HISTory? .....	1275
:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:RX:CURREnt? .....	1276
:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:RX:SECOnds? .....	1277
:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:HISTory? .....	1278
:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:CURREnt? .....	1279
:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:SECOnds? .....	1280
:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:TYPE? .....	1281
:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT .....	1282
:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT? .....	1283

:SOURCE[1..n]:DATA:TELECOM:OPTICAL:ALARM:PORT:LANE .....	1284
:SOURCE[1..n]:DATA:TELECOM:OPTICAL:ALARM:PORT:LANE? .....	1285
:SOURCE[1..n]:DATA:TELECOM:OPTICAL:ALARM:PORT:ALANes .....	1286
:SOURCE[1..n]:DATA:TELECOM:OPTICAL:ALARM:PORT:ALANes? .....	1287
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:UDPProtocol:HISTory? .....	1288
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:UDPProtocol:CURRent? .....	1289
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:UDPProtocol:SECOnds? .....	1290
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:UDPProtocol:COUNt? .....	1291
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:UDPProtocol:RATE? .....	1292
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:IPPRotocol:HISTory? .....	1293
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:IPPRotocol:CURRent? .....	1294
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:IPPRotocol:SECOnds? .....	1295
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:IPPRotocol:COUNt? .....	1296
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:IPPRotocol:RATE? .....	1297
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:HISTory? .....	1298
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:CURRent? .....	1299
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:SECOnds? .....	1300
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:COUNt? .....	1301
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:RATE? .....	1302
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:OVERsize .....	1303
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:OVERsize? .....	1304
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:MANual:TYPE? .....	1305
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:AMOUNt .....	1306
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:AMOUNt? .....	1307
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:INJect .....	1308
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:AUTomated:TYPE? .....	1309
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:AUTomated:RATE .....	1310
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:AUTomated:RATE? .....	1311
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:AUTomated:CONTInuous .....	1312
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:AUTomated:CONTInuous? .....	1313
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:AUTomated .....	1314
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ERRor:MAC:AUTomated? .....	1315
:FETCH[1..n]:DATA:TELECOM:ETHernet:ALARm:PHYsical:GLOBal:HISTory? .....	1316
:FETCH[1..n]:DATA:TELECOM:ETHernet:ALARm:PHYsical:GLOBal:CURRent? .....	1318
:FETCH[1..n]:DATA:TELECOM:ETHernet:ALARm:PHYsical:GLOBal:SECOnds? .....	1320
:FETCH[1..n]:DATA:TELECOM:ETHernet:ALARm:PHYsical:HISTory? .....	1321
:FETCH[1..n]:DATA:TELECOM:ETHernet:ALARm:PHYsical:CURRent? .....	1322
:FETCH[1..n]:DATA:TELECOM:ETHernet:ALARm:PHYsical:SECOnds? .....	1323
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:PHYsical:GLOBal:HISTory? .....	1324
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:PHYsical:GLOBal:CURRent? .....	1325
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:PHYsical:HISTory? .....	1326
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:PHYsical:CURRent? .....	1327
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:PHYsical:SECOnds? .....	1328
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:PHYsical:COUNt? .....	1329
:FETCH[1..n]:DATA:TELECOM:ETHernet:ERRor:PHYsical:RATE? .....	1330
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ALARm:THReshold .....	1331
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ALARm:THReshold? .....	1332
:SOURCE[1..n]:DATA:TELECOM:ETHernet:ALARm:THReshold:DEFault .....	1333

:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE .....	1334
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE? .....	1335
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE .....	1336
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE? .....	1337
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT .....	1338
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT? .....	1339
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:INJect .....	1340
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:TYPE .....	1341
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:TYPE? .....	1342
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:RATE .....	1343
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:RATE? .....	1344
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTInuous .....	1345
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTInuous? .....	1346
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated .....	1347
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated? .....	1348
:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:LINK? .....	1349
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical .....	1350
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical? .....	1351
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:COUNT:TOTAL? .....	1352
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:RATE:TOTAL? .....	1353
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:LANE .....	1354
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:LANE? .....	1355
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:ALANes .....	1356
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:ALANes? .....	1357
:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:GLOBAL:CURRent? .....	1358
:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:GLOBAL:SEConds? .....	1359
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBAL:HISTory? .....	1360
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBAL:CURRent? .....	1361
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBAL:SEConds? .....	1362
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBAL:COUNT? .....	1363
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBAL:RATE? .....	1364
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:MANual:TYPE .....	1365
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:MANual:TYPE? .....	1366
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AMOUNT .....	1367
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AMOUNT? .....	1368
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:INJect .....	1369
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:TYPE .....	1370
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:TYPE? .....	1371
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:RATE .....	1372
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:RATE? .....	1373
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:CONTInuous .....	1374
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:CONTInuous? .....	1375
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated .....	1376
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe .....	1377
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe? .....	1378
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:TYPE .....	1379
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:GLOBAL:HISTory? .....	1380
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:GLOBAL:CURRent? .....	1381

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:HISTory? .....	1382
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:CURRent? .....	1383
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:SECOnds? .....	1384
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:COUnT? .....	1385
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:RATE? .....	1386
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:COUnT:TOTal? .....	1387
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:RATE:TOTal? .....	1388
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:GLOBal:HISTory? .....	1389
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:GLOBal:CURRent? .....	1390
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:GLOBal:SECOnds? .....	1391
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:HISTory? .....	1392
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:CURRent? .....	1394
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:SECOnds? .....	1396
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:HISTory? .....	1397
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:SECOnds? .....	1399
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:CURRent? .....	1402
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:HISTory? .....	1404
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:SECOnds? .....	1407
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:CURRent? .....	1410
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:HISTory? .....	1413
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:SECOnds? .....	1414
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:CURRent? .....	1415
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:COUnT? .....	1416
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:RATE? .....	1417
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:HISTory? .....	1418
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:SECOnds? .....	1419
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:CURRent? .....	1420
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:COUnT? .....	1421
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:RATE? .....	1422
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:HISTory? .....	1423
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:SECOnds? .....	1426
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CURRent? .....	1429
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:HISTory? .....	1432
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:SECOnds? .....	1435
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:CURRent? .....	1438
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:HISTory? .....	1441
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:SECOnds? .....	1443
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CURRent? .....	1445
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:COUnT? .....	1447
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:RATE? .....	1449
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:HISTory? .....	1450
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:SECOnds? .....	1452
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:CURRent? .....	1453
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:COUnT? .....	1455
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:RATE? .....	1456
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:HISTory? .....	1457
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:SECOnds? .....	1459
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CURRent? .....	1461

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:HISTory? .....	1463
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:SEConds? .....	1465
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:CURRent? .....	1467
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:COUnT? .....	1469
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:RATE? .....	1471
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:HISTory? .....	1473
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:SEConds? .....	1474
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:CURRent? .....	1475
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:HISTory? .....	1476
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:SEConds? .....	1477
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent? .....	1478
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:HISTory? .....	1479
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:SEConds? .....	1480
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:CURRent? .....	1481
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:COUnT? .....	1482
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:RATE? .....	1483
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:HISTory? .....	1484
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:SEConds? .....	1486
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent? .....	1487
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:COUnT? .....	1489
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:RATE? .....	1490
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:HISTory? .....	1491
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:CURRent? .....	1492
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:SEConds? .....	1493
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:HISTory? .....	1494
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:CURRent? .....	1495
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:SEConds? .....	1496
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:COUnT? .....	1497
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:RATE? .....	1498
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:HISTory? .....	1499
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:SEConds? .....	1500
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CURRent? .....	1501
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:COUnT? .....	1502
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:RATE? .....	1503
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:MSIM .....	1504
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:MSIM? .....	1505
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE .....	1506
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE? .....	1507
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n] .....	1508
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]? .....	1509
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE .....	1510
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE? .....	1512
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n] .....	1513
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]? .....	1514
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE .....	1515
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE? .....	1516
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOUnt .....	1517
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOUnt? .....	1518

:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:INJect .....	1519
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE .....	1520
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE? .....	1521
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOUnt .....	1522
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOUnt? .....	1523
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:INJect .....	1524
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE .....	1525
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE? .....	1526
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE .....	1527
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE? .....	1528
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated .....	1529
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated? .....	1530
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous .....	1531
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous? .....	1532
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE .....	1533
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE? .....	1534
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE .....	1535
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE? .....	1536
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated .....	1537
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated? .....	1538
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:CONTInuous .....	1539
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:CONTInuous? .....	1540
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:ACHA .....	1541
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:ACHA? .....	1542
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CHAN .....	1543
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CHAN? .....	1544
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE .....	1545
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE? .....	1547
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n] .....	1549
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]? .....	1550
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE .....	1551
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE? .....	1553
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n] .....	1555
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]? .....	1556
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE .....	1557
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE? .....	1558
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOUnt .....	1559
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOUnt? .....	1560
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:INJect .....	1561
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:ACHA .....	1562
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:ACHA? .....	1563
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CHAN .....	1564
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CHAN? .....	1565
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE .....	1566
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE? .....	1567
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUnt .....	1568
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUnt? .....	1569
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:INJect .....	1570



:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE .....	1571
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE? .....	1572
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE .....	1573
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE? .....	1574
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated .....	1575
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated? .....	1576
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous .....	1577
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous? .....	1578
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE .....	1579
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE? .....	1580
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE .....	1581
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE? .....	1582
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated .....	1583
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated? .....	1584
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous .....	1585
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous? .....	1586
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:ACHA .....	1587
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:ACHA? .....	1588
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CHAN .....	1589
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CHAN? .....	1590
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE .....	1591
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE? .....	1592
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n] .....	1593
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]? .....	1594
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:MANual:TYPE .....	1595
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:MANual:TYPE? .....	1596
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AMOUNT .....	1597
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AMOUNT? .....	1598
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:INJect .....	1599
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:TYPE .....	1600
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:TYPE? .....	1601
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:RATE .....	1602
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:RATE? .....	1603
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated .....	1604
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated? .....	1605
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:CONTInuous .....	1606
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:CONTInuous? .....	1607
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:MANual:TYPE .....	1608
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:MANual:TYPE? .....	1609
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOUNT .....	1610
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOUNT? .....	1611
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:INJect .....	1612
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:ACHA .....	1613
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:ACHA? .....	1614
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CHAN .....	1615
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CHAN? .....	1616
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE .....	1617
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE? .....	1618

:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE .....	1619
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE? .....	1620
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated .....	1621
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated? .....	1622
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous .....	1623
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous? .....	1624
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE .....	1625
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE? .....	1626
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n] .....	1627
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]? .....	1628
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:TYPE .....	1629
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:TYPE? .....	1630
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n] .....	1631
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]? .....	1632
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE .....	1633
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE? .....	1634
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOUNT .....	1635
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOUNT? .....	1636
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:INJect .....	1637
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TCM[1..n]:TYPE .....	1638
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TCM[1..n]:TYPE? .....	1639
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AMOUNT .....	1640
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AMOUNT? .....	1641
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:INJect .....	1642
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:TYPE .....	1643
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:TYPE? .....	1644
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:TYPE .....	1645
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:TYPE? .....	1646
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:RATE .....	1647
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:RATE? .....	1648
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:RATE .....	1649
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:RATE? .....	1650
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AUTomated .....	1651
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AUTomated? .....	1652
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AUTomated .....	1653
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AUTomated? .....	1654
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous .....	1655
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous? .....	1656
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:CONTInuous .....	1657
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:CONTInuous? .....	1658
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm:TYPE .....	1659
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm:TYPE? .....	1660
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm .....	1661
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm? .....	1662
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALARm:TYPE .....	1663
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALARm:TYPE? .....	1664
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALARm .....	1665
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALARm? .....	1666

:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:MANual:TYPE .....	1667
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:MANual:TYPE? .....	1668
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AMOUNT .....	1669
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AMOUNT? .....	1670
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:INJect .....	1671
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:TYPE .....	1672
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:TYPE? .....	1673
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:RATE .....	1674
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:RATE? .....	1675
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:CONTInuous .....	1676
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:CONTInuous? .....	1677
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated .....	1678
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated? .....	1679
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:LANE .....	1680
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:LANE? .....	1681
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALANes .....	1682
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALANes? .....	1683
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:TYPE .....	1684
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:TYPE? .....	1685
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:AMOUNT .....	1686
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:AMOUNT? .....	1687
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:INJect .....	1688
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:TYPE .....	1689
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:TYPE? .....	1690
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:RATE .....	1691
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:RATE? .....	1692
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:CONTInuous .....	1693
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:CONTInuous? .....	1694
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated .....	1695
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated? .....	1696
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:TYPE .....	1697
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:TYPE? .....	1698
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:AMOUNT .....	1699
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:AMOUNT? .....	1700
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:INJect .....	1701
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:TYPE .....	1702
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:TYPE? .....	1703
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:RATE .....	1704
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:RATE? .....	1705
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:CONTInuous .....	1706
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:CONTInuous? .....	1707
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated .....	1708
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated? .....	1709
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:LANE .....	1710
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:LANE? .....	1711
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:ALANes .....	1712
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:ALANes? .....	1713
:FETCh[1..n]:DATA:TELEcom:EOTN:XCODIng:TRANscode:RX:ALARm:HISTory? .....	1714

:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ALARm:CURRent?	1715
:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ALARm:SEConds?	1716
:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ERRor:HISTory?	1717
:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ERRor:CURRent?	1718
:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ERRor:SEConds?	1719
:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ERRor:COUNT?	1720
:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ERRor:RATE?	1721
:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:HISTory?	1722
:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:CURRent?	1724
:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:SEConds?	1725
:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:COUNT?	1726
:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE?	1727
:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:COUNT:TOTAL?	1728
:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE:TOTAL?	1729
:FETCh[1..n]:DATA:TELEcom:ETHernet:EOTN:ALARm:LINK?	1730
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE?	1731
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E	1732
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E?	1733
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:HISTory?	1734
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:SEConds?	1735
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:CURRent?	1736
:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead	1737
:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:HISTory?	1738
:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:CURRent?	1739
:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:SEConds?	1740
:FETCh[1..n]:DATA:TELEcom:ALARm:HISTory?	1741
:FETCh[1..n]:DATA:TELEcom:ALARm:SEConds?	1743
:FETCh[1..n]:DATA:TELEcom:ALARm:CURRent?	1745
:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?	1747
:FETCh[1..n]:DATA:TELEcom:ERRor:SEConds?	1749
:FETCh[1..n]:DATA:TELEcom:ERRor:CURRent?	1751
:FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?	1753
:FETCh[1..n]:DATA:TELEcom:ERRor:RATE?	1755
:FETCh[1..n]:DATA:TELEcom:ERRor:COUNT:TOTAL?	1757
:FETCh[1..n]:DATA:TELEcom:ERRor:RATE:TOTAL?	1758
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE	1759
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE?	1760
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AMOUnt	1761
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AMOUnt?	1762
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:INJect	1763
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:TYPE	1764
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:TYPE?	1765
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:RATE	1766
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:RATE?	1767
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated	1768
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated?	1769
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:CONTInuous	1770
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:CONTInuous?	1771

:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:HISTory?	1772
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:CURRent?	1773
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:SEConds?	1774
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:HISTory?	1775
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:CURRent?	1776
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:SEConds?	1777
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:COUNt?	1778
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:RATE?	1779
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:TYPE	1780
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:TYPE?	1781
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:RATE	1782
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:RATE?	1783
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTInuous	1784
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTInuous?	1785
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated	1786
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated?	1787
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:MANual:TYPE	1788
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:MANual:TYPE?	1789
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AMOUNt	1790
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AMOUNt?	1791
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:INJect	1792
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:HISTory?	1793
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:CURRent?	1795
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:SEConds?	1797
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:HISTory?	1798
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:CURRent?	1800
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:SEConds?	1802
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:COUNt?	1803
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:RATE?	1804
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:TYPE	1805
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:TYPE?	1806
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe	1807
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe?	1808
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:TYPE	1809
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:TYPE?	1810
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:PERIOD	1811
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:PERIOD?	1812
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI	1813
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI?	1814
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel	1815
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel?	1816
:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:CMF	1817
:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:CMF?	1818
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:HISTory?	1819
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:SEConds?	1820
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:CURRent?	1821
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:HISTory?	1822
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:SEConds?	1823

:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:CURRent?	1824
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:COUNt?	1825
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:RATE?	1826
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:HISTory?	1827
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:SEcSnds?	1828
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:CURRent?	1829
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:HISTory?	1830
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:SEcSnds?	1831
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:CURRent?	1832
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:COUNt?	1833
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:RATE?	1834
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:HISTory?	1835
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:SEcSnds?	1836
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:CURRent?	1837
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:COUNt?	1838
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:RATE?	1839
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:HISTory?	1840
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:SEcSnds?	1842
:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:CURRent?	1844
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:TYPE	1846
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:TYPE?	1847
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon	1848
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon?	1849
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:MANual:TYPE	1850
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:MANual:TYPE?	1851
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:AMOUnt	1852
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:AMOUnt?	1853
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:INJect	1854
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:AUTomated:TYPE	1855
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:AUTomated:TYPE?	1856
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:AUTomated:RATE	1857
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:AUTomated:RATE?	1858
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:AUTomated	1859
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:AUTomated?	1860
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:AUTomated:CONTInuous	1861
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:AUTomated:CONTInuous?	1862
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:BURSt:TYPE	1863
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:BURSt:TYPE?	1864
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:BURSt:MODE	1865
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:BURSt:MODE?	1866
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:BURSt:DURation	1867
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:BURSt:DURation?	1868
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:BURSt:PERIOD	1869
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:BURSt:PERIOD?	1870
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:BURSt	1871
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTIon:BURSt?	1872
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt:TYPE	1873
:SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt:TYPE?	1874

:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt:MODE .....	1875
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt:MODE? .....	1876
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt:DURation .....	1877
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt:DURation? .....	1878
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt:PERIOD .....	1879
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt:PERIOD? .....	1880
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt .....	1881
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt? .....	1882
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:TYPE .....	1883
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:TYPE? .....	1884
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE .....	1885
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE? .....	1886
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:MANual:TYPE .....	1887
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:MANual:TYPE? .....	1888
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AMOUnt .....	1889
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AMOUnt? .....	1890
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:INJect .....	1891
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AUTomated:TYPE .....	1892
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AUTomated:TYPE? .....	1893
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AUTomated:RATE .....	1894
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AUTomated:RATE? .....	1895
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AUTomated .....	1896
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AUTomated? .....	1897
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AUTomated:CONTInuous .....	1898
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AUTomated:CONTInuous? .....	1899
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:TYPE .....	1900
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:TYPE? .....	1901
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:MODE .....	1902
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:MODE? .....	1903
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:DURation .....	1904
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:DURation? .....	1905
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:PERIOD .....	1906
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:PERIOD? .....	1907
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt .....	1908
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt? .....	1909
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:TYPE .....	1910
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:TYPE? .....	1911
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:MODE .....	1912
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:MODE? .....	1913
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:DURation .....	1914
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:DURation? .....	1915
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:PERIOD .....	1916
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:PERIOD? .....	1917
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt .....	1918
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt? .....	1919
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:TYPE .....	1920
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:TYPE? .....	1922
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH .....	1923

:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH? .....	1924
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:MANual:TYPE .....	1925
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:MANual:TYPE? .....	1926
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:AMOUNT .....	1927
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:AMOUNT? .....	1928
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:INJECT .....	1929
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:AUTomated:TYPE .....	1930
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:AUTomated:TYPE? .....	1931
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:AUTomated:RATE .....	1932
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:AUTomated:RATE? .....	1933
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:AUTomated .....	1934
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:AUTomated? .....	1935
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:AUTomated:CONTInuous .....	1936
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:AUTomated:CONTInuous? .....	1937
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:BURSt:TYPE .....	1938
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:BURSt:TYPE? .....	1939
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:BURSt:MODE .....	1940
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:BURSt:MODE? .....	1941
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:BURSt:DURation .....	1942
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:BURSt:DURation? .....	1943
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:BURSt:PERiod .....	1944
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:BURSt:PERiod? .....	1945
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:BURSt .....	1946
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:PATH:BURSt? .....	1947
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:TYPE .....	1948
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:TYPE? .....	1949
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:MODE .....	1950
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:MODE? .....	1951
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:DURation .....	1952
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:DURation? .....	1953
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:PERiod .....	1954
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:PERiod? .....	1955
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt .....	1956
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt? .....	1957
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:MANual:TYPE .....	1958
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:MANual:TYPE? .....	1959
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:MANual:AMOUNT .....	1960
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:MANual:AMOUNT? .....	1961
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:INJECT .....	1962
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:AUTomated:TYPE .....	1963
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:AUTomated:TYPE? .....	1964
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:AUTomated:RATE .....	1965
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:AUTomated:RATE? .....	1966
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:AUTomated .....	1967
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:AUTomated? .....	1968
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:AUTomated:CONTInuous .....	1969
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:AUTomated:CONTInuous? .....	1970
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM:TYPE .....	1971



:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM:TYPE?	1972
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM	1973
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM?	1974
:FETCH[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:HISTory?	1975
:FETCH[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:SEcS?	1977
:FETCH[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:CURRent?	1978
:FETCH[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:COUnT?	1980
:FETCH[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:RAte?	1981
:FETCH[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM:HISTory?	1982
:FETCH[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM:SEcS?	1984
:FETCH[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM:CURRent?	1985
:FETCH[1..n]:DATA:TELEcom:CAU:ALARm:GLOBal:SEcS?	1987
:FETCH[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:SEcS?	1988
:FETCH[1..n]:DATA:TELEcom:CAU:ALARm:GLOBal:TX:STATus?	1989
<b>Traces - SONET/SDH</b>	<b>1990</b>
:SENSe[1..n]:DATA:TELEcom:SDHSONet:PATH:OVERhead:TIM:COpy	1990
:FETCH[1..n]:DATA:TELEcom:SDHSONet:POVerhead:J[1..n]:TIM:PAATern:RECEived?	1991
:SENSe[1..n]:DATA:TELEcom:SDHSONet:SECTIon:OVERhead:TIM:COpy	1992
:FETCH[1..n]:DATA:TELEcom:SDHSONet:SOVerhead:J[1..n]:TIM:PAATern:RECEived?	1993
:FETCH[1..n]:DATA:TELEcom:SDHSONet:HOP:TCAPident:N[1..n]:RECEived?	1994
:SENSe[1..n]:DATA:TELEcom:SDHSONet:HOP:TCAPident:COpy	1995
<b>Traces - OTN</b>	<b>1996</b>
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?	1996
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:B?	1997
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:OPSPec:B?	1998
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:B?	1999
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:B?	2000
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:OPSPec:B?	2001
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:B?	2002
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:B?	2003
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:OPSPec:B?	2004
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:B?	2005
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:B?	2006
:FETCH[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:OPSPec:B?	2007
:FETCH[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:B?	2008
:FETCH[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:B?	2009
:FETCH[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:OPSPec:B?	2010
:FETCH[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:B?	2011
:FETCH[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:B?	2012
:FETCH[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:OPSPec:B?	2013
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:COpy	2014
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:COpy	2015
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:COpy	2016
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:COpy	2017
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:COpy	2018
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:COpy	2019
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:COpy	2020
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:COpy	2021

:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:COPY .....	2022
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:COPY .....	2023
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:COPY .....	2024
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:COPY .....	2025
<b>Logger and Alarms/Errors Logger .....</b>	<b>2026</b>
:FETCh[1..n]:DATA:TELEcom:LOGGer:EVENTs? .....	2026
:FETCh[1..n]:DATA:TELEcom:LOGGer:LIST? .....	2027
<b>Performance Monitoring .....</b>	<b>2028</b>
:FETCh[1..n]:DATA:TELEcom:PATtern:PM:STATistics? .....	2028
:FETCh[1..n]:DATA:TELEcom:SONet:SECTion:PM:STATistics? .....	2030
:FETCh[1..n]:DATA:TELEcom:SONet:LINE:PM:STATistics? .....	2033
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:PM:STATistics? .....	2036
<b>Traffic - Ethernet .....</b>	<b>2039</b>
:SENSE[1..n]:DATA:TELEcom:ETHernet:PACKet:LINE:UTILization? .....	2039
:SENSE[1..n]:DATA:TELEcom:ETHernet:PACKet:BANDwidth? .....	2040
:SENSE[1..n]:DATA:TELEcom:ETHernet:PACKet:FRAME:RATE? .....	2041
:SENSE[1..n]:DATA:TELEcom:ETHernet:FRAME:COUNt:RX? .....	2042
:SOURce[1..n]:DATA:TELEcom:ETHernet:FRAME:COUNt:TX? .....	2043
:SENSE[1..n]:DATA:TELEcom:ETHernet:FSIZE:COUNt? .....	2044
:SENSE[1..n]:DATA:TELEcom:ETHernet:FSIZE:PERCentAge? .....	2046
:SENSE[1..n]:DATA:TELEcom:ETHernet:PACKet:FRAME:COUNt? .....	2047
<b>Traffic - Flow Control .....</b>	<b>2048</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES? .....	2048
:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:ABORt? .....	2049
:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:RX? .....	2050
:SOURce[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSE:TIME .....	2051
:SOURce[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSE:TIME? .....	2052
:SOURce[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSE:INJect .....	2053
:SOURce[1..n]:DATA:TELEcom:ETHernet:PINJectiON:MAC:DESTinatiON:ADDReSS .....	2054
:SOURce[1..n]:DATA:TELEcom:ETHernet:PINJectiON:MAC:DESTinatiON:ADDReSS? .....	2055
:SOURce[1..n]:DATA:TELEcom:ETHernet:PINJectiON:MAC:DESTinatiON:ADDReSS:ENABle .....	2056
:SOURce[1..n]:DATA:TELEcom:ETHernet:PINJectiON:MAC:DESTinatiON:ADDReSS:ENABle? .....	2057
:FETCh[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSE:TIME:RX? .....	2058
:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:TX? .....	2059
<b>FTFL/PT .....</b>	<b>2060</b>
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDicatiON? .....	2060
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE? .....	2061
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier? .....	2062
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:SPECific? .....	2063
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDicatiON? .....	2064
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE? .....	2065
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENtifier? .....	2066
:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:SPECific? .....	2067
:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE:RECEived? .....	2068
:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe:RECEived? .....	2069
:SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:COPY .....	2070
<b>OTL-SDT .....</b>	<b>2071</b>

:FETCh[1..n]:DATA:TELEcom:OTL:SDT:STATistics?	2071
:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LONGest?	2072
:FETCh[1..n]:DATA:TELEcom:OTL:SDT:SHORtest?	2073
:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LAST?	2074
:FETCh[1..n]:DATA:TELEcom:OTL:SDT:AVERAge?	2075
:FETCh[1..n]:DATA:TELEcom:OTL:SDT:TOTAl?	2076
:FETCh[1..n]:DATA:TELEcom:OTL:SDT:COUNt?	2077
:FETCh[1..n]:DATA:TELEcom:OTL:SDT:DEFect?	2078
:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LONGest:DISRUption:DURation?	2079
:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LONGest:DISRUption:LANE?	2080
:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LANE:DISRUption?	2081
<b>GFP-F/GFP-T</b>	<b>2082</b>
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:COUNt:TX?	2082
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:TX?	2084
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:BANDwith:TX?	2086
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:EFFiciency:TX?	2087
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:COUNt:RX?	2088
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:RX?	2090
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:BANDwith:RX?	2092
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:EFFiciency:RX?	2093
:FETCh[1..n]:DATA:TELEcom:GFP:SUPerblock:COUNt:TX?	2094
:FETCh[1..n]:DATA:TELEcom:GFP:SUPerblock:RATE:TX?	2095
:FETCh[1..n]:DATA:TELEcom:GFP:SUPerblock:COUNt:RX?	2096
:FETCh[1..n]:DATA:TELEcom:GFP:SUPerblock:RATE:RX?	2097
:SENSE[1..n]:DATA:TELEcom:GFP:FRAMe:MISMATCH:COUNt?	2098
:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel:MISMATCH:COUNt?	2099
<b>Streams - Throughput</b>	<b>2100</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:MAXimum?	2100
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:MINimum?	2101
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:CURRent?	2102
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:AVERAge?	2103
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:TOTAl:RXRate?	2104
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:AVERAge:VERDICT?	2105
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:CURRent:VERDICT?	2106
<b>Streams - Jitter</b>	<b>2107</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MINimum?	2107
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MAXimum?	2108
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:AVERAge?	2109
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:CURRent?	2110
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:ESTimate?	2111
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MAXimum:VERDICT?	2112
<b>Streams - Latency</b>	<b>2113</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MAXimum?	2113
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MINimum?	2114
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:AVERAge?	2115
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:CURRent?	2116
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MAXimum:VERDICT?	2117
<b>Streams - Frame Loss / Out-of-Sequence</b>	<b>2118</b>

:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:HISTory?	2118
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:CURRent?	2119
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:SECOnds?	2120
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:COUNt?	2121
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:RATE?	2122
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:OOSequence:VERDict?	2123
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:FLOSS:VERDict?	2125
<b>Service Configuration - Ramp</b>	<b>2127</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSS:VERDict?	2127
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter:VERDict?	2130
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency:VERDict?	2132
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate:VERDict?	2134
<b>Service Configuration - Burst</b>	<b>2136</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOSS?	2136
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter?	2138
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency?	2140
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate?	2142
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:TXRate?	2144
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSS?	2147
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter?	2150
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency?	2153
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate?	2156
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOSS:VERDict?	2159
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter:VERDict?	2161
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency:VERDict?	2163
:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate:VERDict?	2165
<b>Labels</b>	<b>2167</b>
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:PATH:LABel?	2167
<b>MPLS</b>	<b>2168</b>
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:BANdwidth?	2168
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:RATE?	2169
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:UTILization?	2170
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:COUNt?	2171
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:BANdwidth?	2172
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:RATE?	2173
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:UTILization?	2174
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:COUNt?	2175
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:FRAMes:TX?	2176
:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:FRAMes:RX?	2177
<b>SDT (Multi-Channel OTN)</b>	<b>2178</b>
:FETCh[1..n]:DATA:TELEcom:SDT:LOTImestamp?	2178
:FETCh[1..n]:DATA:TELEcom:SDT:LOCHannel?	2179
:FETCh[1..n]:DATA:TELEcom:SDT:LATIImestamp?	2180
:FETCh[1..n]:DATA:TELEcom:SDT:LACHannel?	2181
:FETCh[1..n]:DATA:TELEcom:SDT:CHDIruption?	2182
:FETCh[1..n]:DATA:TELEcom:SDT:CHAThreshold?	2183

:FETCh[1..n]:DATA:TELEcom:SDT:CHMOnitored? .....	2184
<b>APS .....</b>	<b>2185</b>
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE .....	2185
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE? .....	2186
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest .....	2187
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest? .....	2189
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest .....	2191
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest? .....	2193
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel .....	2195
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel? .....	2196
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel .....	2197
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel? .....	2198
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture .....	2199
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture? .....	2200
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:OMODE .....	2201
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:OMODE? .....	2202
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE .....	2203
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE? .....	2204
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:DNODE .....	2205
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:DNODE? .....	2206
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE .....	2207
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE? .....	2208
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest .....	2209
:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest? .....	2210
:SENSe[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE .....	2211
:SENSe[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE? .....	2212
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest? .....	2213
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest? .....	2214
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel? .....	2215
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel? .....	2216
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture? .....	2217
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:OMODE? .....	2218
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE? .....	2219
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:DNODE? .....	2220
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE? .....	2221
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest? .....	2222
<b>OH - OTN .....</b>	<b>2223</b>
:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead .....	2223
:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead? .....	2226
:SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead? .....	2228
:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead .....	2230
:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead? .....	2231
:SENSe[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead? .....	2233
:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead .....	2235
:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead? .....	2236
:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead? .....	2238
:SOURce[1..n]:DATA:TELEcom:OTN:OH:REStore:DEFault .....	2241
:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead .....	2242

:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead? .....	2244
:SENSE[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead? .....	2246
:SOURCE[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead .....	2248
:SOURCE[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead? .....	2250
:SOURCE[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:DEFault .....	2252
:SENSE[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead? .....	2253
:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI .....	2255
:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI? .....	2256
:SENSE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI? .....	2257
:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead? .....	2258
:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:DEFault .....	2260
:SENSE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead? .....	2261
:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI .....	2264
:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI? .....	2265
:SENSE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI? .....	2266
<b>OH - SONET/SDH .....</b>	<b>2267</b>
:SOURCE[1..n]:DATA:TELEcom:SONet:OH:SECTIon:OVERhead .....	2267
:SOURCE[1..n]:DATA:TELEcom:SONet:OH:SECTIon:OVERhead? .....	2269
:SOURCE[1..n]:DATA:TELEcom:SONet:OH:SECTIon:OVERhead:DEFault .....	2271
:SENSE[1..n]:DATA:TELEcom:SONet:OH:SECTIon:OVERhead? .....	2272
:SOURCE[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead .....	2274
:SOURCE[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead? .....	2276
:SOURCE[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead:DEFault .....	2278
:SENSE[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead? .....	2279
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n] .....	2281
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]? .....	2283
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]:DEFault .....	2285
:SENSE[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]? .....	2286
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n] .....	2288
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]? .....	2290
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]:DEFault .....	2292
:SENSE[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]? .....	2293
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead .....	2295
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead? .....	2297
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead:DEFault .....	2299
:SENSE[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead? .....	2300
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:REStore:DEFault .....	2302
<b>OH - GFP-F/GFP-T .....</b>	<b>2303</b>
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI .....	2303
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI? .....	2304
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI .....	2306
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI? .....	2307
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI .....	2309
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI? .....	2310
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI .....	2312
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI? .....	2313
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID .....	2315
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID? .....	2316

:SOURCE[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe .....	2318
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe? .....	2319
:FETCh[1..n]:DATA:TELEcom:GFP:OH:DFRames? .....	2321
:FETCh[1..n]:DATA:TELEcom:GFP:OH:MFRames? .....	2323
:FETCh[1..n]:DATA:TELEcom:GFP:OH:RPTiframes? .....	2325
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:REStore:DEFault .....	2327
:SOURCE[1..n]:DATA:TELEcom:GFP:OH:DEFault .....	2328
<b>RTD .....</b>	<b>2329</b>
:SENSe[1..n]:DATA:TELEcom:RTD:MODE .....	2329
:SENSe[1..n]:DATA:TELEcom:RTD:MODE? .....	2330
:SENSe[1..n]:DATA:TELEcom:RTD .....	2331
:SENSe[1..n]:DATA:TELEcom:RTD? .....	2332
:FETCh[1..n]:DATA:TELEcom:RTD:DELay:STATus? .....	2333
:FETCh[1..n]:DATA:TELEcom:RTD:DELay:LAST? .....	2334
:FETCh[1..n]:DATA:TELEcom:RTD:DELay:MAXimum? .....	2335
:FETCh[1..n]:DATA:TELEcom:RTD:DELay:MINimum? .....	2336
:FETCh[1..n]:DATA:TELEcom:RTD:DELay:AVERage? .....	2337
:FETCh[1..n]:DATA:TELEcom:RTD:COUNt:SUCCEssful? .....	2338
:FETCh[1..n]:DATA:TELEcom:RTD:COUNt:FAILEd? .....	2339
:SENSe[1..n]:DATA:TELEcom:RTD:RESEt .....	2340
<b>Pointer Adjustment .....</b>	<b>2341</b>
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:VALue? .....	2341
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:INCRement:SIZE .....	2342
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:INCRement:SIZE? .....	2343
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:DECRement:SIZE .....	2344
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:DECRement:SIZE? .....	2345
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:INCRement .....	2346
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:DECRement .....	2347
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:NEW:VALue .....	2348
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:NEW:VALue? .....	2349
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:NEW .....	2350
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:NEW:FLAG .....	2351
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:NEW:FLAG? .....	2352
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:VALue? .....	2353
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:OFFSEt? .....	2354
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:INCRement:COUNt? .....	2355
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:DECRement:COUNt? .....	2356
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:INCRement:SECOnds? .....	2357
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:DECRement:SECOnds? .....	2358
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:NDF:COUNt? .....	2359
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:NNDF:COUNt? .....	2360
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:NDF:SECOnds? .....	2361
:FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:POINter:NNDF:SECOnds? .....	2362
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:PATTern .....	2363
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:PATTern? .....	2365
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:TYPE .....	2367
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:TYPE? .....	2368
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:PERiodic:STATus .....	2369

:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:PERiodic:STATus?	2370
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:INITcool:STATus	2371
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:INITcool:STATus?	2372
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:TIMeline:VALue	2373
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:TIMeline:VALue?	2374
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter	2376
:SOURCE[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter?	2377
:FETCh[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:VALue?	2378
:FETCh[1..n]:DATA:TELEcom:SDHSONet:SEQUence:POINter:STATus?	2379
<b>40/100G Advanced - Lanes Mapping &amp; Skew</b>	<b>2380</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:RESet	2380
:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPing:DEFault	2381
:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPing:RANDom	2382
:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPing:MANual	2383
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:THReshold	2384
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:THReshold?	2385
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:THReshold:DEFault	2386
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:RESet	2387
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:MAPPing:DEFault	2388
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:MAPPing:RANDom	2389
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:MAPPing:MANual	2390
:SENSe[1..n]:DATA:TELEcom:OTN:OTL:RX?	2391
<b>40/100G Advanced - CFP/CFP2/CFP4/QSFP Control</b>	<b>2392</b>
:SOURCE[1..n]:DATA:TELEcom:CFP:CSETting	2392
:SOURCE[1..n]:DATA:TELEcom:CFP:CSETting?	2393
:SOURCE[1..n]:DATA:TELEcom:CFP:CPRating	2395
:SOURCE[1..n]:DATA:TELEcom:CFP:CPRating?	2396
:SOURCE[1..n]:DATA:TELEcom:CFP:STATus?	2397
:SENSe[1..n]:DATA:TELEcom:CFP:TX:STATus?	2399
:SOURCE[1..n]:DATA:TELEcom:CFP:RCLock	2400
:SOURCE[1..n]:DATA:TELEcom:CFP:RCLock?	2401
:SOURCE[1..n]:DATA:TELEcom:MDIO:ADDRess	2402
:SOURCE[1..n]:DATA:TELEcom:MDIO:ADDRess?	2403
:SOURCE[1..n]:DATA:TELEcom:MDIO:DATA	2404
:SOURCE[1..n]:DATA:TELEcom:MDIO:DATA?	2405
:SOURCE[1..n]:DATA:TELEcom:MDIO:READ	2406
:SOURCE[1..n]:DATA:TELEcom:MDIO:WRITe	2407
:SOURCE[1..n]:DATA:TELEcom:MDIO:STARt:ADDRess	2408
:SOURCE[1..n]:DATA:TELEcom:MDIO:STARt:ADDRess?	2409
:SOURCE[1..n]:DATA:TELEcom:MDIO:END:ADDRess	2410
:SOURCE[1..n]:DATA:TELEcom:MDIO:END:ADDRess?	2411
<b>40/100G Advanced - Pre-Emphasis</b>	<b>2412</b>
:SOURCE[1..n]:DATA:TELEcom:CONFig	2412
:SOURCE[1..n]:DATA:TELEcom:CONFig?	2413
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:VOD	2414
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:VOD?	2415
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:PRET	2416
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:PRET?	2417



:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:POST[1..n] .....	2418
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:POST[1..n]? .....	2419
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APR .....	2420
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APR? .....	2421
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APOS[1..n] .....	2422
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APOS[1..n]? .....	2423
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:ECONtrol .....	2424
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:ECONtrol? .....	2425
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:EGain .....	2426
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:EGain? .....	2427
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:AVOD .....	2428
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:AVOD? .....	2429
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:AECControl .....	2430
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:AECControl? .....	2431
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:AEGain .....	2432
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:AEGain? .....	2433
:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RESet .....	2434
<b>Default/Random/Manual Mapping .....</b>	<b>2435</b>
:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:TX .....	2435
:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:TX? .....	2436
:SENSe[1..n]:DATA:TELEcom:ETHernet:LLAYer:RX? .....	2438
:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane .....	2439
:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane? .....	2440
:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX .....	2441
:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX? .....	2442
<b>Reset/Manual Skew .....</b>	<b>2443</b>
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane .....	2443
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane? .....	2444
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX .....	2445
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX? .....	2446
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:TX .....	2447
:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:TX? .....	2448
<b>Bulk Read .....</b>	<b>2449</b>
:SOURCE[1..n]:DATA:TELEcom:MDIO:BULK:READ .....	2449
:FETCh[1..n]:DATA:TELEcom:MDIO:BULK:READ:INFormation? .....	2450
<b>Ping &amp; Trace Route .....</b>	<b>2451</b>
:FETCh[1..n]:DATA:TELEcom:PING:STATistics:TX? .....	2451
:FETCh[1..n]:DATA:TELEcom:PING:STATistics:RX? .....	2452
:FETCh[1..n]:DATA:TELEcom:PING:STATistics:LOST? .....	2453
:FETCh[1..n]:DATA:TELEcom:PING:STATistics:MINimum? .....	2454
:FETCh[1..n]:DATA:TELEcom:PING:STATistics:MAXimum? .....	2455
:FETCh[1..n]:DATA:TELEcom:PING:STATistics:AVERAge? .....	2456
:FETCh[1..n]:DATA:TELEcom:PING:STATistics:RESults? .....	2457
:SOURCE[1..n]:DATA:TELEcom:PING:SETup:RUN .....	2458
:SOURCE[1..n]:DATA:TELEcom:PING:SETup:RUN? .....	2459
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TOUT .....	2460
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TOUT? .....	2461
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:DELay .....	2462

:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:DELay?	2463
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:DSIZe	2464
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:DSIZe?	2465
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TTL	2466
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TTL?	2467
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TOS	2468
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TOS?	2469
:FETCh[1..n]:DATA:TELEcom:TRACe:STATistics:TX?	2470
:FETCh[1..n]:DATA:TELEcom:TRACe:STATistics:RX?	2471
:FETCh[1..n]:DATA:TELEcom:TRACe:STATistics:RESults?	2472
:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:RUN	2473
:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:RUN?	2474
:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:TOUT	2475
:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:TOUT?	2476
:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:HCOunt	2477
:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:HCOunt?	2478
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:CONtinuous	2479
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:CONtinuous?	2480
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ATTempts	2481
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ATTempts?	2482
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDReSS:SOURce:IP?	2483
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDReSS:DESTination:IP	2484
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDReSS:DESTination:IP?	2485
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDReSS:DESTination:IP:USTReam	2486
:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDReSS:DESTination:IP:USTReam?	2487
<b>Filters</b>	<b>2488</b>
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer	2488
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer?	2489
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE	2490
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE?	2492
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP	2494
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP?	2495
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FRAMe:FORMat	2496
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FRAMe:FORMat?	2497
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC	2498
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC?	2499
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP	2500
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?	2501
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices	2503
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices?	2504
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence	2506
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence?	2507
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP	2509
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP?	2510
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC	2511
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC?	2512
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:UDP	2513
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:UDP?	2514

---

:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS .....	2516
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS? .....	2517
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLan:ID .....	2519
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLan:ID? .....	2521
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLan:PRIOriTy .....	2523
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLan:PRIOriTy? .....	2525
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRoTocol .....	2527
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRoTocol? .....	2528
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHerTyPe .....	2530
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHerTyPe? .....	2531
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:DESTInatIon:IP .....	2532
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:DESTInatIon:IP? .....	2533
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:DESTInatIon:MAC .....	2534
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:DESTInatIon:MAC? .....	2535
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:DESTInatIon:UDP .....	2536
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:DESTInatIon:UDP? .....	2537
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:DSERvices .....	2538
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:DSERvices? .....	2539
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:PRECEdence .....	2541
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:PRECEdence? .....	2542
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:SOURce:IP .....	2544
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:SOURce:IP? .....	2545
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:SOURce:MAC .....	2546
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:SOURce:MAC? .....	2547
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:SOURce:UDP .....	2548
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:SOURce:UDP? .....	2549
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:TOS .....	2550
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:TOS? .....	2551
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:VLan:ID .....	2553
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:VLan:ID? .....	2554
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:VLan:PRIOriTy .....	2556
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:VLan:PRIOriTy? .....	2558
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:IPPRoTocol .....	2560
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:IPPRoTocol? .....	2561
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:ETHerTyPe .....	2562
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASk:ETHerTyPe? .....	2563
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ENABled:TIME? .....	2564
:SENSE[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:BANdWIdth? .....	2565
:SENSE[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:RATE? .....	2566
:SENSE[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:UTILIZatIon? .....	2567
:SENSE[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:COUNt? .....	2568
:SENSE[1..n]:DATA:TELEcom:ETHernet:FILTer:STATISTICS? .....	2569
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator .....	2571
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator? .....	2572
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT .....	2573
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT? .....	2574
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACkET:OPEN .....	2575
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACkET:OPEN? .....	2576

:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLoSe .....	2577
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLoSe? .....	2578
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DEStination:IPVerSion .....	2579
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DEStination:IPVerSion? .....	2580
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPVerSion .....	2581
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPVerSion? .....	2582
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLABel:IPVerSion .....	2583
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLABel:IPVerSion? .....	2584
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHEader:IPVerSion .....	2586
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHEader:IPVerSion? .....	2587
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPVerSion .....	2589
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPVerSion? .....	2590
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence:IPVerSion .....	2591
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence:IPVerSion? .....	2592
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS:IPVerSion .....	2593
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS:IPVerSion? .....	2594
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DEStination:IPVerSion .....	2595
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DEStination:IPVerSion? .....	2596
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IPVerSion .....	2597
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IPVerSion? .....	2598
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:FLABel:IPVerSion .....	2599
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:FLABel:IPVerSion? .....	2600
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHEader:IPVerSion .....	2602
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHEader:IPVerSion? .....	2603
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices:IPVerSion .....	2605
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices:IPVerSion? .....	2606
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECEdence:IPVerSion .....	2608
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECEdence:IPVerSion? .....	2609
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS:IPVerSion .....	2611
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS:IPVerSion? .....	2612
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DEStination:IPV .....	2613
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DEStination:IPV? .....	2614
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERvices:IPV .....	2615
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERvices:IPV? .....	2616
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECEdence:IPV .....	2617
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECEdence:IPV? .....	2618
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:NHEader:IPV .....	2619
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:NHEader:IPV? .....	2620
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:FLABel:IPV .....	2621
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:FLABel:IPV? .....	2622
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IPV .....	2623
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IPV? .....	2624
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS:IPV .....	2625
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS:IPV? .....	2626
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DEStination:IPV .....	2627
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DEStination:IPV? .....	2628
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERvices:IPV .....	2629
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERvices:IPV? .....	2630

:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PREcedence:IPV .....	2631
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PREcedence:IPV? .....	2632
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:NHEader:IPV .....	2633
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:NHEader:IPV? .....	2634
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:FLABel:IPV .....	2636
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:FLABel:IPV? .....	2637
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IPV .....	2639
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IPV? .....	2640
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS:IPV .....	2641
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS:IPV? .....	2642
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MLABel[1..n] .....	2643
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MLABel[1..n]? .....	2644
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MCOS[1..n] .....	2646
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MCOS[1..n]? .....	2647
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel[1..n] .....	2649
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel[1..n]? .....	2650
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS[1..n] .....	2652
:SENSE[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS[1..n]? .....	2653
<b>Packet Capture .....</b>	<b>2655</b>
:SOURce[1..n]:DATA:TELEcom:CAPTure:FILTer:TYPE .....	2655
:SOURce[1..n]:DATA:TELEcom:CAPTure:FILTer:TYPE? .....	2656
:SOURce[1..n]:DATA:TELEcom:CAPTure:FRAMe:SIZE .....	2657
:SOURce[1..n]:DATA:TELEcom:CAPTure:FRAMe:SIZE? .....	2658
:SOURce[1..n]:DATA:TELEcom:CAPTure:BYTE .....	2659
:SOURce[1..n]:DATA:TELEcom:CAPTure:BYTE? .....	2660
:SOURce[1..n]:DATA:TELEcom:CAPTure:TRIGger .....	2661
:SOURce[1..n]:DATA:TELEcom:CAPTure:TRIGger? .....	2662
:SOURce[1..n]:DATA:TELEcom:CAPTure:TSource .....	2663
:SOURce[1..n]:DATA:TELEcom:CAPTure:TSource? .....	2664
:SOURce[1..n]:DATA:TELEcom:CAPTure:TSource:TYPE .....	2665
:SOURce[1..n]:DATA:TELEcom:CAPTure:TSource:TYPE? .....	2666
:SOURce[1..n]:DATA:TELEcom:ETHernet:GLOBal:CONTRol .....	2667
:SOURce[1..n]:DATA:TELEcom:ETHernet:GLOBal:CONTRol? .....	2668
:FETCh[1..n]:DATA:TELEcom:ETHernet:STATus? .....	2669
:FETCh[1..n]:DATA:TELEcom:ETHernet:BUFFer:UTILization? .....	2670
:FETCh[1..n]:DATA:TELEcom:ETHernet:CFG:STATus? .....	2671
:FETCh[1..n]:DATA:TELEcom:ETHernet:FRAMe:COUNt? .....	2672
:FETCh[1..n]:DATA:TELEcom:ETHernet:TRIGger:ERRor? .....	2673
<b>Triggered Frame Details .....</b>	<b>2674</b>
:SENSE[1..n]:DATA:TELEcom:CAPTure:TSource:FNUMber? .....	2674
:SENSE[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:MAC? .....	2675
:SENSE[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:IP? .....	2676
:SENSE[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:PORT? .....	2677
:SENSE[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:MAC? .....	2678
:SENSE[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:IP? .....	2679
:SENSE[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:PORT? .....	2680
<b>Filter Configuration .....</b>	<b>2681</b>
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TYPE .....	2681

:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TYPE? .....	2683
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IP .....	2684
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IP? .....	2685
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:MAC .....	2686
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:MAC? .....	2687
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:UDP .....	2688
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:UDP? .....	2689
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERvices .....	2691
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERvices? .....	2692
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PREcedence .....	2693
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PREcedence? .....	2694
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IP .....	2695
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IP? .....	2696
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:MAC .....	2697
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:MAC? .....	2698
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:UDP .....	2699
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:UDP? .....	2700
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS .....	2701
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS? .....	2702
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:ID .....	2703
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:ID? .....	2704
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:PRIority .....	2706
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:PRIority? .....	2707
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:IPPRotocol .....	2709
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:IPPRotocol? .....	2710
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:ETHertype .....	2711
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:ETHertype? .....	2712
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IP .....	2713
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IP? .....	2714
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:MAC .....	2715
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:MAC? .....	2716
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:UDP .....	2717
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:UDP? .....	2718
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERvices .....	2719
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERvices? .....	2720
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PREcedence .....	2721
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PREcedence? .....	2722
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IP .....	2723
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IP? .....	2724
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:MAC .....	2725
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:MAC? .....	2726
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:UDP .....	2727
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:UDP? .....	2728
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS .....	2729
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS? .....	2730
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:ID .....	2731
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:ID? .....	2732
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:PRIority .....	2733

:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:PRiority? .....	2734
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:IPPRotocol .....	2736
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:IPPRotocol? .....	2737
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:ETHertype .....	2738
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:ETHertype? .....	2739
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator .....	2740
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator? .....	2741
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator:NOT .....	2742
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator:NOT? .....	2743
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:OPEN .....	2744
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:OPEN? .....	2745
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:CLoSe .....	2746
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:CLoSe? .....	2747
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MLABel[1..n] .....	2748
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MLABel[1..n]? .....	2749
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MCOS[1..n] .....	2750
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MCOS[1..n]? .....	2751
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MLABel[1..n] .....	2752
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MLABel[1..n]? .....	2753
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MCOS[1..n] .....	2754
:SENSE[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MCOS[1..n]? .....	2755
<b>GMP .....</b>	<b>2756</b>
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CMStatus? .....	2756
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CNDStatus? .....	2758
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:CMStatus? .....	2759
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:CNDStatus? .....	2760
<b>Client Offset .....</b>	<b>2761</b>
:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency? .....	2761
:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:OFFSet .....	2762
:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:OFFSet? .....	2763
:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:ENABle .....	2764
:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:ENABle? .....	2765
:SENSE[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:EFRequency? .....	2766
:SENSE[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:FOANalysis:ENABle .....	2767
:SENSE[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:FOANalysis:ENABle? .....	2768
:SENSE[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency? .....	2769
<b>Traffic Scan .....</b>	<b>2771</b>
:SOURce[1..n]:DATA:TELEcom:TSCan:SCAN:ENABle .....	2771
:SOURce[1..n]:DATA:TELEcom:TSCan:SCAN:ENABle? .....	2772
:SOURce[1..n]:DATA:TELEcom:TSCan:LEVel:TYPE .....	2773
:SOURce[1..n]:DATA:TELEcom:TSCan:LEVel:TYPE? .....	2774
:FETCh[1..n]:DATA:TELEcom:TSCan:LINK:RATE? .....	2775
:FETCh[1..n]:DATA:TELEcom:TSCan:DISCOvered? .....	2776
:FETCh[1..n]:DATA:TELEcom:TSCan:LRReached:STATus? .....	2777
:FETCh[1..n]:DATA:TELEcom:TSCan:LIST? .....	2778
:FETCh[1..n]:DATA:TELEcom:TSCan:STATistics:FCOunt:TOTal? .....	2779
:FETCh[1..n]:DATA:TELEcom:TSCan:STATistics:RATE:TOTal? .....	2780
<b>Discover Remote Button .....</b>	<b>2781</b>

---

:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:CONNect .....	2781
:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:CONNect? .....	2782
:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:DISConnect .....	2783
:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:SSUBnet .....	2784
:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:SSUBnet? .....	2785
:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:SCANtarget:TYPE .....	2786
:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:SCANtarget:TYPE? .....	2787
:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:LOOP:UP .....	2788
:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:LOOP:DOWN .....	2789
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:CStatus .....	2790
:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:CStatus? .....	2791
:FETCh[1..n]:DATA:TELEcom:ETHernet:DUALtest:STATistics? .....	2792
:FETCh[1..n]:DATA:TELEcom:ETHernet:REMote:RSCStatus? .....	2793
<b>Lpbk Tool Button (Interface) .....</b>	<b>2794</b>
:FETCh[1..n]:DATA:TELEcom:MODule:DETail:MiD? .....	2794
:FETCh[1..n]:DATA:TELEcom:MODule:DETail:SNUMber? .....	2795
:FETCh[1..n]:DATA:TELEcom:MODule:DETail:AHRevision? .....	2796

**10 Obsolete SCPI Commands ..... 2797**



# **1** *Introducing the FTB/IQS-85100G SCPI Commands*

The FTB/IQS-85100G of modules can be remotely controlled using SCPI commands. You can also use these commands directly on the unit to build scripts to automate test processes.

### Conventions

Before using the product described in this guide, you should understand the following conventions:



#### **WARNING**

Indicates a potentially hazardous situation which, if not avoided, could result in *death or serious injury*. Do not proceed unless you understand and meet the required conditions.



#### **CAUTION**

Indicates a potentially hazardous situation which, if not avoided, may result in *minor or moderate injury*. Do not proceed unless you understand and meet the required conditions.



#### **CAUTION**

Indicates a potentially hazardous situation which, if not avoided, may result in *component damage*. Do not proceed unless you understand and meet the required conditions.



#### **IMPORTANT**

Refers to information about this product you should not overlook.

## 2 *Getting Started*

This section contains a brief summary of the FTB/IQS-85100G specific commands. Following steps will give an idea about the command and sequence of commands to execute and perform a specific test.

*To create a BERT test for example:*

**1. \*CLS**

This command clears the register of module.

**2. INST:CAT:FULL?**

This command will detect the module attached to the platform back panel and return the name of the module and its position with Unit number and Slot number.

For example "FTB-88xxx-Power Blazer", 10. This information is added with LINS keyword before each and every SCPI command. Following SCPI commands describes the use of LINS.

**3. LINS10:SOURce:DATA:TELecom:TEST:TYPE EBERT**

This command selects Ethernet Bert Test.

**4. LINS10:SOURce:DATA:TELecom:ITYPE LANE4X10**

This command sets the interface type as 4X10.

**5. LINS10:SOURce:DATA:TELecom:ITYPE?**

This query returns the interface type as 4X10.

**6. LINS10:SOURce:DATA:TELecom:ETHernet:PORT:TRANsceiver CFP**

This command selects the type of connector as CFP.

**7. LINS10:SOURce:DATA:TELecom:ETHernet:PORT:TRANsceiver?**

This query returns the type of connector.

**8. LINS10:SOURce:DATA:TELecom:ETHernet:BERT:FRAMing FRAMEDLAYER2**

This command selects the framing type as FRAMEDLAYER2.

**9. LINS10:SOURce:DATA:TELEcom:ETHernet:BERT:FRAMing?**

This query returns the framing type.

**10. LINS10:SENSe:DATA:TELEcom:ALASer ON**

This command enables or disables the status of all lasers.

**11. LINS10:SENSe:DATA:TELEcom:ALASer?**

This query returns the current state of all lasers.

**12. LINS10:SOURce:DATA:TELEcom:PATtern:TYPE PRBs2E9**

This command selects the payload pattern type for the transmitter as PRBS2E9.

**13. LINS10:SOURce:DATA:TELEcom:PATtern:TYPE?**

This query returns the payload pattern type of the transmitter.

**14. LINS10:SOURce:DATA:TELEcom:TEST ON**

This command starts the manual test.

**15. LINS10:SOURce:DATA:TELEcom:TEST?**

This query returns the status of the manual test.

**16. LINS10:SOURce:DATA:TELEcom:PATtern:ERRor:PATtern:MANual: TYPE BIT**

This command sets the type of pattern error as BIT.

**17. LINS10:SOURce:DATA:TELEcom:PATtern:ERRor:PATtern:MANual: TYPE?**

This query returns the type of pattern error.

**18. LINS10:SOURce:DATA:TELEcom:PATtern:ERRor:PATtern:AMOUNT 25**

This command sets the amount of pattern error to be injected into the instrument as 25.

**19. LINS10:SOURce:DATA:TELEcom:PATtern:ERRor:PATtern:AMOUNT?**

This query returns the amount of pattern error injected into the instrument.

**20. LINS10:SOURce:DATA:TELEcom:PATtern:ERRor:PATtern:INJect**

This command injects the type of pattern error.

**21. LINS10:SOURce:DATA:TELEcom:TEST OFF**

This command stops the manual test.

**Note:** *In User Interface, if "--" is displayed for any field, the related SCPI results will be according to the following table conditions:*

Data Type	Has Minimum Value	Command Result
< NR1 NUMERIC RESPONSE DATA > < NR2 NUMERIC RESPONSE DATA > < NR3 NUMERIC RESPONSE DATA > < HEXADECIMAL NUMERIC RESPONSE DATA >	Yes	[Minimum value]
< NR1 NUMERIC RESPONSE DATA > < NR2 NUMERIC RESPONSE DATA > < NR3 NUMERIC RESPONSE DATA > < HEXADECIMAL NUMERIC RESPONSE DATA >	No	This data does not yet have a value, it is pending.
< STRING RESPONSE DATA > < CHARACTER RESPONSE DATA >	No	This data does not yet have a value, it is pending.



# 3 *Communicating Through TCP/IP over Telnet*

## Introducing TCP/IP over Telnet

The EXFO Instrument Control provides SCPI automation or remote control over Telnet through TCP/IP as a Windows Service that continuously listens to a port from a Telnet server (FTB/IQS/LTB) on which modules to be tested are connected.

TCP/IP protocols are used for communication.

*Note: Port 5024 is designated for sending SCPI commands in the Telnet protocol.*

Most of Windows versions include the Telnet client and the Telnet server components. With these components, you can create a remote command console session on a remote computer.

Commands can be executed simply by logging on the server using the Telnet interface.

There are two types of commands that can be sent over Telnet: SCPI commands and internal protocol commands of the TCP/IP over Telnet service. The internal commands allow you to perform certain actions such as send SCPI commands as a script instead of one by one, force the disconnection of an active session, view the status of modules and of connected clients, etc.

## Features

- A client from any operating system (Windows, Linux, or Unix) can use the freely available Telnet components to connect to the service.
- A client can connect to multiple modules at a time.
- A user can connect to multiple modules through single/multiple sessions.
- A client can execute single commands or a batch of commands.
- A user can disconnect any client/session that is already connected.

## Configuring Your Unit and Modules to Work With TCP/IP over Telnet

The TCP/IP over Telnet Service, which is part of the EXFO Instrument Control, is a mediator between the Telnet client and the test instrument.

Once your unit is configured properly, any request from the Telnet client is transferred to the appropriate instrument. The instrument executes the request and returns the response to TCP/IP over Telnet accordingly.



### IMPORTANT

If you are working with an FTB-2/ FTB-2 Pro, an FTB-4 Pro, an LTB-1, or an LTB-8 unit, before being able to control instruments with SCPI commands, you must first allow remote access to these instruments.



## Communicating Through TCP/IP over Telnet

### *Configuring Your Unit and Modules to Work With TCP/IP over Telnet*

**To allow remote access to your instruments (FTB-2/FTB-2 Pro, FTB-4 Pro, LTB-1, and LTB-8 only):**

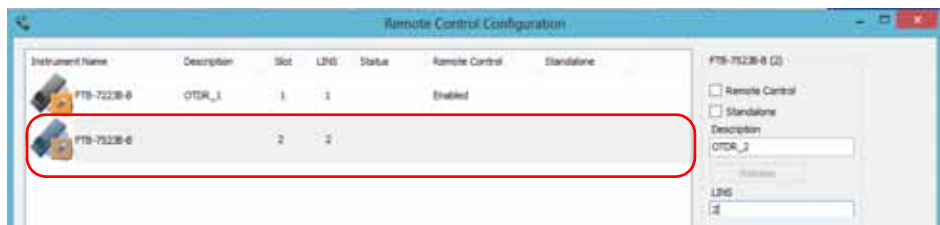
1. From ToolBox X, tap or click the **System Settings** button.
2. Tap or click **Remote Control Configuration**.



3. If necessary, tap or click **Change settings**, and then, when the application prompts you to authorize the changes to your unit (identified as “computer”), select **Yes**.



4. From the **Remote Control Configuration** window you will see all the inserted modules. Select the module for which you want to have remote access.



## Communicating Through TCP/IP over Telnet

### *Configuring Your Unit and Modules to Work With TCP/IP over Telnet*

---

5. Set the parameters:
  - Select **Remote Control** to be able to access the module remotely (via TCP/IP over Telnet or other).
  - Select **Standalone** to leave the module active even if all users close their dedicated applications.
6. If desired, under **Description**, type a description that will help you identify the instrument.



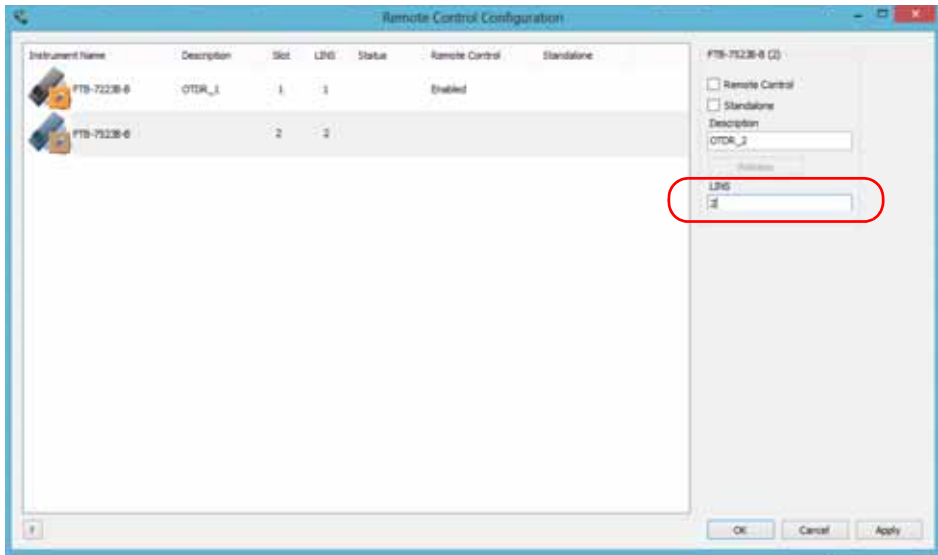
**Note:** *You can enter up to 10 characters. The description can correspond to the test interface ID or to any other short text of your choice.*

## Communicating Through TCP/IP over Telnet

### Configuring Your Unit and Modules to Work With TCP/IP over Telnet

---

7. If necessary, under **LINS**, modify the logical instrument number that you will use to access the instrument remotely.



**Note:** If the **LINS** column is empty, it means that the corresponding module cannot be controlled using SCPI commands.

8. Tap or click **Apply** to confirm your changes or **OK** to apply your changes and close the window.

**Note:** This information will be updated the next time you start the module application, or set Instrument Control in remote mode. Refer to the corresponding module documentation for more details.

# Communicating Through TCP/IP over Telnet

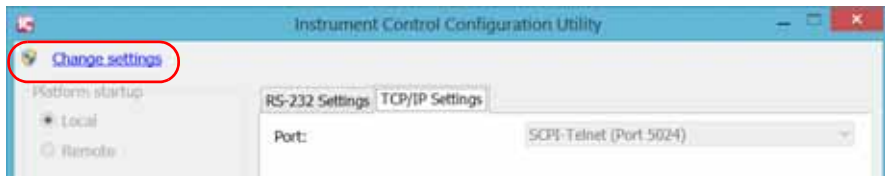
*Configuring Your Unit and Modules to Work With TCP/IP over Telnet*

---

**To activate TCP/IP over Telnet on your unit:**

1. Access the Instrument Control Configuration utility:
  - On an IQS-600: From IQS Manager, click the **Utilities** function tab, and then click **Instrument Control Configuration**.
  - On an FTB-500: From ToolBox, tap the **System Settings** button, then tap **Instrument Control Configuration**.
  - On an FTB-2/FTB-2 Pro, an FTB-4 Pro, an LTB-1, or an LTB-8: From ToolBox X, tap or click the **System Settings** button, then tap or click **Instrument Control Configuration**.
2. If necessary, tap or click **Change settings**, and then, when the application prompts you to authorize the changes to your unit, select **Yes**.

**IQS-600,  
FTB-500,  
LTB-1, and  
LTB-8**



**FTB-2/FTB-2  
Pro, and  
FTB-4 Pro**



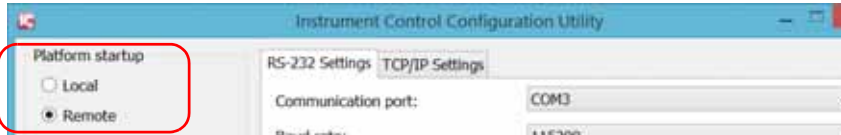
## Communicating Through TCP/IP over Telnet

### Configuring Your Unit and Modules to Work With TCP/IP over Telnet

---

#### 3. Allow remote control:

- On an IQS-600, FTB-500, LTB-1, or LTB-8: Under **Platform Startup**, select **Remote**.



- On an FTB-2/FTB-2 Pro, or FTB-4 Pro: Select the **Allow automation on instruments** check box.



#### 4. Under **Communication Type**, select **TCP/IP**.



- 5. Tap or click **Apply**, and then **OK**.
- 6. Depending on the unit you are using, restart either IQS Manager, ToolBox, or ToolBox X.

## Executing SCPI Commands Over Telnet

You can remotely control the modules by executing SCPI commands through TCP/IP over Telnet. The commands are sent remotely from the Telnet client (on a computer) to the Telnet server (in this case, the IQS, FTB, or LTB unit).


To execute a single SCPI command, you can type or paste the command directly in the Telnet editor window.

To execute multiple SCPI commands (script), you must enclose them within a BEGIN and END block in the Telnet editor window.

You can connect from a remote Windows client or a Linux (or Unix) remote client.

**Note:** *The Telnet client is available on the FTB-500, FTB-2 running Windows 10 IoT Enterprise, FTB-2 Pro, FTB-4 Pro, LTB-8, and IQS-600 controller if you intend to use these units as computers to connect to a Telnet server. However, on an FTB-2 running Windows Embedded 8 Standard, or on an LTB-1, the Telnet client is not available. You must use the PuTTY application to establish communication.*

**To connect through TCP/IP over Telnet from a remote Windows client:**

1. From your computer, start Windows.
2. On the taskbar, click **Start** (Start button (  ) under Windows 8.1 and Windows 10) and select **Run**.

**Note:** *Depending on the operating system, Run can sometimes be found under Windows System.*

3. In the **Open** box, type *telnet*, and then click **OK**.



**Note:** *If you receive an error message, it probably means that the Telnet client is not already activated on your computer. In this case, in the **Open** box, type `pkgmgr /iu:TelnetClient`, and then click **OK** to enable the client. Once it is done, perform step 3 again.*

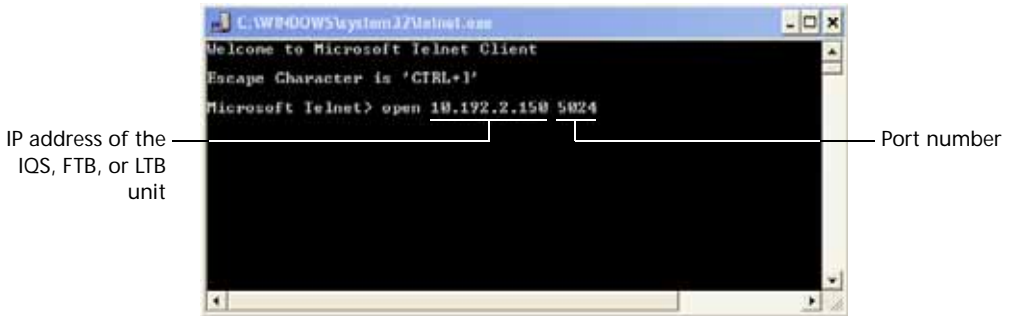
## Communicating Through TCP/IP over Telnet

### Executing SCPI Commands Over Telnet

---

4. In the displayed Telnet editor window, type the `OPEN <IP_ADDRESS_OF_TELNET_SERVER> <PORT>` command to connect to the TCP/IP Telnet Service.

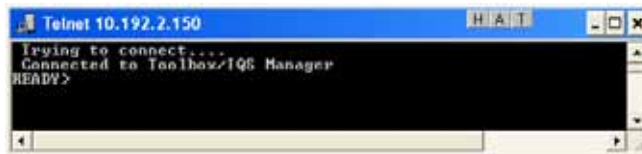
Example: `open 10.192.2.45 5024`



**Note:** Port 5024 is designated for sending SCPI commands in the Telnet protocol.

5. Press ENTER to establish a connection with the Service.

Once the connection is established, the `READY>` prompt is displayed in the Telnet editor window.



**Note:** If the connection cannot be established, the **Connection to host lost** message is displayed instead.



6. Enter the desired SCPI commands as follows:

- For a single SCPI command: Type or copy the desired command in the Telnet editor window, and then press ENTER to execute it.

```

Telnet 10.192.2.150
Trying to connect...
Connected to Testbox/IQE Manager
READY> inst:cat:full17
Transport Blazer (10.7Ch/2) IQE-8130NGE".13,"Packet Blazer IQE-8518B".14
READY>
    
```

- For multiple SCPI commands: Copy the desired commands from any script file, enclose them in a BEGIN and END block in the Telnet editor window, and then press ENTER. For more information, see *Internal Commands of the TCP/IP over Telnet Protocol* on page 21.

```

READY> BEGIN
LINS13:SOURCE:DATA:TELEcom:CLEAr
LINS13:OUTPut:TELEcom:CONNector?
LINS13:OUTPut:TELEcom:CONNector: OPTical
LINS13:OUTPut:TELEcom:CONNector?
LINS13:SOURCE:DATA:TELEcom:INTERface:TYPE?
LINS13:SOURCE:DATA:TELEcom:INTERface:TYPE? OC3
LINS13:SOURCE:DATA:TELEcom:INTERface:TYPE?
LINS13:SOURCE:DATA:TELEcom:HOP:TYPE?
LINS13:SOURCE:DATA:TELEcom:HOP:TYPE STS1
LINS13:SOURCE:DATA:TELEcom:HOP:TYPE?
LINS13:SOURCE:DATA:TELEcom:LOP:TYPE UT15
LINS13:SOURCE:DATA:TELEcom:LOP:TYPE?
END
Previous test cleared successfully
OPTICAL
Command executed successfully
OPTICAL
NONE
Command executed successfully
OC3
NONE
Command executed successfully
STS1
Command executed successfully
UT15
READY>
    
```

**Note:** *Multiple commands that are not enclosed in BEGIN...END blocks will be executed, but problems may occur (results of one command mixed with the results of another one, skipped commands, etc.).*

Once at least one valid command is executed, the module is blocked for any other sessions until the module is released. For more information, see *Releasing Modules* on page 20.

7. Click  to close the session.

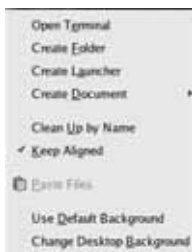
## Communicating Through TCP/IP over Telnet

### Executing SCPI Commands Over Telnet

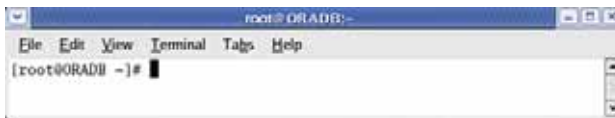
---

*To connect through TCP/IP over Telnet from a remote Linux client:*

1. From your computer, right-click on the desktop, and then click **Open Terminal**.



The command prompt is displayed in the Telnet editor window.

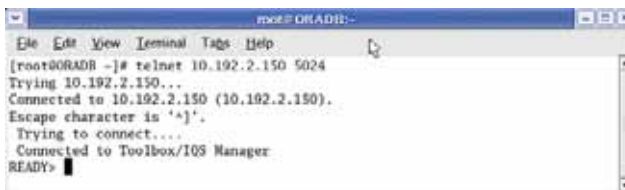


2. Connect to the TCP/IP Telnet Service by typing the *OPEN <IP\_ADDRESS\_OF\_TELNET\_SERVER> <PORT>* command:

Example: *open 10.192.2.45 5024*

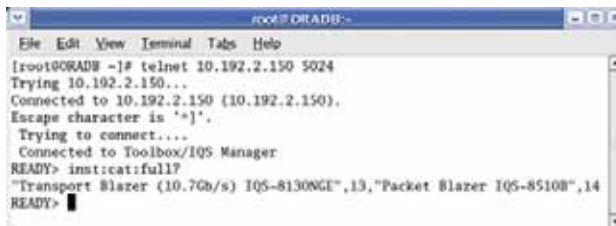
**Note:** *Port 5024 is designated for sending SCPI commands in the Telnet protocol.*

The connection is established when the message **Connected to Toolbox/IQS Manager** is displayed in the Telnet editor window.



3. Enter the desired SCPI commands as follows:

- For a single SCPI command: Type or copy the desired command in the Telnet editor window, and then press ENTER to execute it.



```
root@ORADB:~# telnet 10.192.2.150 5024
Trying 10.192.2.150...
Connected to 10.192.2.150 (10.192.2.150).
Escape character is '^)'.
Trying to connect....
Connected to Toolbox/IQS Manager
READY> imst:cat:full?
Transport Blazer (10.7Gb/s) IQS-8130NGE",13,"Packet Blazer IQS-8510B",14
READY> █
```

- For multiple SCPI commands: Copy the desired commands from any script file, enclose them in a BEGIN and END block in the Telnet editor window, and then press ENTER. For more information, see *Internal Commands of the TCP/IP over Telnet Protocol* on page 21.

**Note:** *Multiple commands that are not enclosed in BEGIN...END blocks will be executed, but problems may occur (results of one command mixed with the results of another one, skipped commands, etc.).*

Once at least one valid command is executed, the module is blocked for any other sessions until the module is released. For more information, see *Releasing Modules* on page 20.

4. Click  to close the session.

## Releasing Modules

A module is blocked as soon as a connection is established from any user session executing the internal command `CONNECT LINS`, or a valid instrument (SCPI) command.

For example, when the SCPI command `LINS10:SOURce:DATA:TELEcom:CLear` command is executed for the first time by client session `10.192.2.155:1364`, the module is blocked for any other client/session until you release it.

A module is released by one of the following actions:

- Executing the `CLOSE LINS` command to disconnect the link with the module. For more information, see *CLOSE LINS* on page 23.
- Executing the `CLOSE` command to end the current session once the execution of all the desired commands has been completed. For more information, see *CLOSE* on page 23.
- Closing the current session by clicking the Close button on the Telnet editor windows' title bar.
- Shutting down and restarting the client computer.
- A network interruption.

A module can also be released when you terminate the communication by using the `KILL LINS` command. For more information, see *KILL LINS* on page 25.

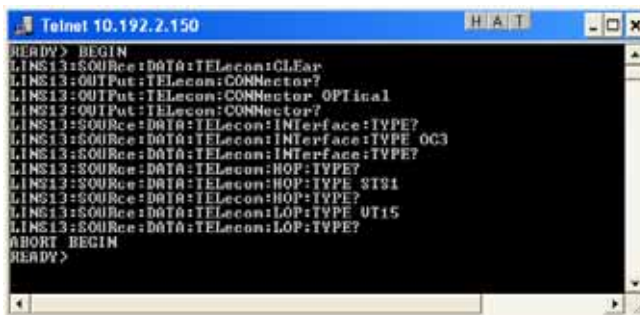
## Internal Commands of the TCP/IP over Telnet Protocol

The internal commands allow you to perform certain actions such as send SCPI commands as a script instead of one by one, force the disconnection of an active session, view the status of modules and of connected clients, etc. The internal commands are not case-sensitive.

### ABORT BEGIN

Syntax: ABORT BEGIN

The ABORT BEGIN command stops the execution of the SCPI commands that are enclosed in a BEGIN and END block, and returns to the READY> prompt in the Telnet editor window.



```
Telnet 10.192.2.150  H A T
READY> BEGIN
LINS13:SOURCE:DATA:TELEcon:CLear
LINS13:OUTPut:TELEcon:CONNector?
LINS13:OUTPut:TELEcon:CONNector? OPTinal
LINS13:OUTPut:TELEcon:CONNector?
LINS13:SOURCE:DATA:TELEcon:INTERface:TYPE?
LINS13:SOURCE:DATA:TELEcon:INTERface:TYPE? OC3
LINS13:SOURCE:DATA:TELEcon:INTERface:TYPE?
LINS13:SOURCE:DATA:TELEcon:HOP:TYPE?
LINS13:SOURCE:DATA:TELEcon:HOP:TYPE? ST31
LINS13:SOURCE:DATA:TELEcon:HOP:TYPE?
LINS13:SOURCE:DATA:TELEcon:LOP:TYPE? UT15
LINS13:SOURCE:DATA:TELEcon:LOP:TYPE?
ABORT BEGIN
END
READY>
```

## Communicating Through TCP/IP over Telnet

### *Internal Commands of the TCP/IP over Telnet Protocol*

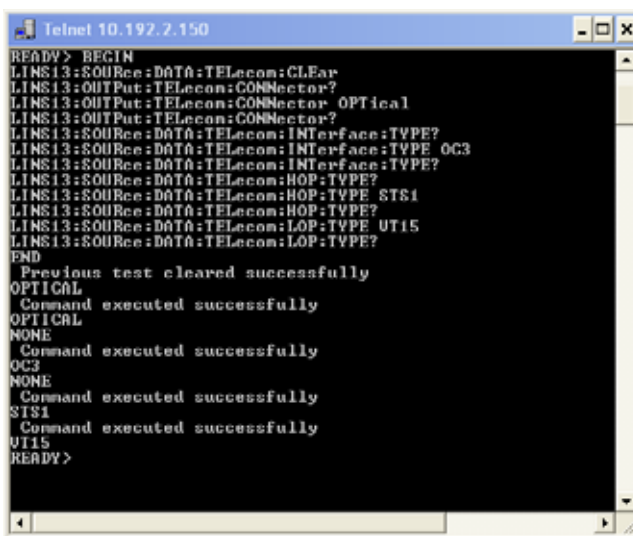
---

## BEGIN and END

To execute multiple SCPI commands, you must enclose them in BEGIN and END blocks in a Telnet editor window.

**Note:** *To execute a single command, simply type or paste the command in the Telnet editor window.*

**Note:** *You cannot enclose internal commands in a BEGIN and END block, except the ABORT BEGIN command.*



```
Telnet 10.192.7.150
READY> BEGIN
LINS13:SOURce:DATA:TELEcon:CLEar
LINS13:OUTPut:TELEcon:CONNector?
LINS13:OUTPut:TELEcon:CONNector OPTical
LINS13:OUTPut:TELEcon:CONNector?
LINS13:SOURce:DATA:TELEcon:INTErface:TVPE?
LINS13:SOURce:DATA:TELEcon:INTErface:TVPE OC3
LINS13:SOURce:DATA:TELEcon:INTErface:TVPE?
LINS13:SOURce:DATA:TELEcon:HOP:TVPE?
LINS13:SOURce:DATA:TELEcon:HOP:TVPE SIS1
LINS13:SOURce:DATA:TELEcon:HOP:TVPE?
LINS13:SOURce:DATA:TELEcon:LOP:TVPE UT15
LINS13:SOURce:DATA:TELEcon:LOP:TVPE?
END
Previous test cleared successfully
OPTICAL
Command executed successfully
OPTICAL
NONE
Command executed successfully
OC3
NONE
Command executed successfully
SIS1
Command executed successfully
UT15
READY>
```

## CLOSE

Syntax: CLOSE

The CLOSE command terminates the current Telnet session.



```
Telnet 10.192.2.150
READY> CLOSE
Connection to host lost.
Press any key to continue...
```

## CLOSE LINS

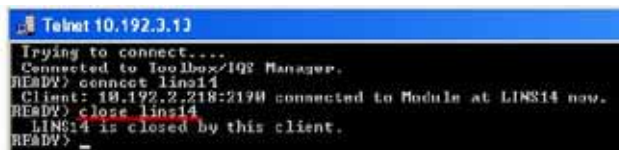
Syntax: CLOSE LINS< Unit\_Number> < Module\_Slot\_Number>

You must specify the unit number and the slot number identifying the module for which you want to close the connections.

This command allows to close active connections. You can send this command to close all client's connections with any module, including the current connection.

If the command is not executed successfully, the possible reasons could be:

- The module is not present at the specified position.
- The provided information does not correspond to a valid LINS.



```
Telnet 10.192.3.13
Trying to connect...
Connected to Toolbox/IGP Manager.
READY> connect lins14
Client: 10.192.2.218:2190 connected to Module at LINS14 now.
READY> close lins14
LINS14 is closed by this client.
READY>
```

## Communicating Through TCP/IP over Telnet

### *Internal Commands of the TCP/IP over Telnet Protocol*

---

## CONNECT LINS

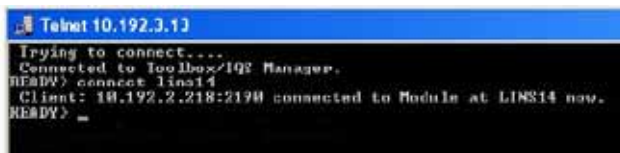
Syntax: `CONNECT LINS< Unit_Number> < Module_Slot_Number>`

You must specify the unit number and the slot number identifying the module to which the session will connect.

This command allows to connect to different modules through TCP/IP. You can connect to multiple modules from a single session.

If the command is not executed successfully, the possible reasons could be:

- The module is already connected to a different client session.
- The module is not present at the specified position.
- The provided information does not correspond to a valid LINS.



```
Telnet 10.192.3.13
Trying to connect...
Connected to Toolbox/IQE Manager.
READY> connect lins14
Client: 10.192.2.218:2190 connected to Module at LINS14 now.
READY> =
```

**Note:** *For backward compatibility reasons, to connect to a single module, you do not have to use the `CONNECT LINS` command. A valid instrument command (for example, `Lins10:SOURce:DATA:TELEcom:CLEar`) for a valid LINS position will work for a first module. However, you will need to use the `CONNECT LINS` command if you want to connect to other modules as well.*




## KILL LINS

Syntax: KILL LINS< Unit\_Number> < Module\_Slot\_Number>

You must specify the unit number and the slot number identifying the module for which you want to terminate the session.

This command allows any user to terminate the session that contains the specified connection (LINS). This means it will terminate all active connections that belongs to a session.



```
Telnet 10.192.2.150
READY> kill lins13
This client session is terminated
READY>
```

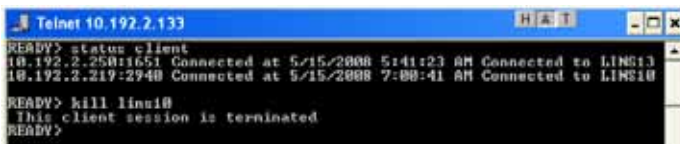
To know the status of the modules before terminating connections using the KILL LINS command, you can first enter the STATUS CLIENT command. For more information, see *STATUS CLIENT* on page 26.

In the example below, two modules are connected: LINS13 and LINS10.



```
Telnet 10.192.2.133
READY> status client
10.192.2.250:1651 Connected at 5/15/2008 5:41:23 AM Connected to LINS13
10.192.2.219:2905 Connected at 5/15/2008 6:04:42 AM Connected to LINS10
READY>
```

To disconnect the LINS10 module used by another session, enter the *kill lins10* command. The **This client session is terminated** message is displayed once the module is disconnected.



```
Telnet 10.192.2.133
READY> status client
10.192.2.250:1651 Connected at 5/15/2008 5:41:23 AM Connected to LINS13
10.192.2.219:2940 Connected at 5/15/2008 7:00:41 AM Connected to LINS10
READY> kill lins10
This client session is terminated
READY>
```

## Communicating Through TCP/IP over Telnet

### *Internal Commands of the TCP/IP over Telnet Protocol*

---

Enter again the STATUS CLIENT command to confirm the termination of the module (LINS10 in our example). Only the information of the remaining connected client is displayed.

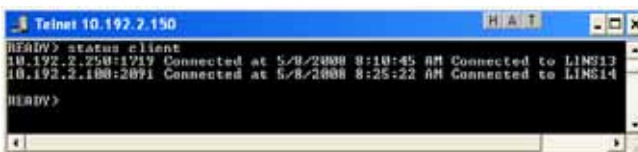


```
Telnet 10.192.2.133
READY> status client
10.192.2.258:1651 Connected at 5/15/2008 5:41:23 AM Connected to LINS13
```

## STATUS CLIENT

Syntax: STATUS CLIENT

This command lists out all clients with their connection time and modules.

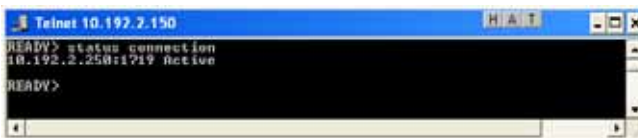


```
Telnet 10.192.2.150
READY> status client
10.192.2.250:1719 Connected at 5/8/2008 8:10:45 AM Connected to LINS13
10.192.2.188:2091 Connected at 5/8/2008 8:25:22 AM Connected to LINS14
READY>
```

## STATUS CONNECTION

Syntax: STATUS CONNECTION

This command lists out all the connections with their *Idle* or *Active* status.



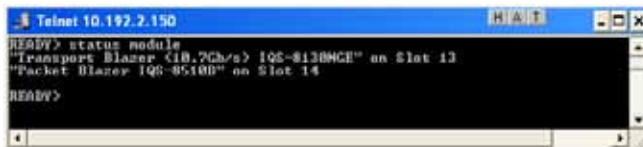
```
Telnet 10.192.2.150
READY> status connection
10.192.2.250:1719 Active
READY>
```

**Note:** *If any connection is idle for a certain period (5 minutes by default), the service automatically changes the status to Idle.*

## STATUS MODULE

Syntax: STATUS MODULE

This command lists out all the modules with the slot numbers where they are located.

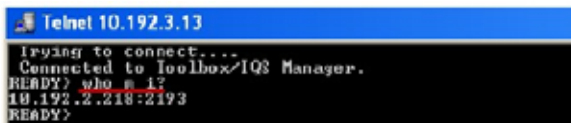


```
Telnet 10.192.2.150
READY> status module
"Transport Blazer (10.7Gh/s) IQS-8130MGE" on Slot 13
"Packet Blazer IQS-05100" on Slot 14
READY>
```

## WHO M I?

Syntax: WHO M I?

This command retrieves the IP address and the communication port of the current session.



```
Telnet 10.192.3.13
Trying to connect...
Connected to Toolbox/IQS Manager.
READY> who m i?
10.192.2.218:2193
READY>
```



# 4 *SCPI Command List - General*

## Delay Between SCPI Commands

*:CONFig[1..n]:WAIT:TIME*

## Test Applications

*:SOURce[1..n]:DATA:TELEcom:TEST:TYPE*  
*:SOURce[1..n]:DATA:TELEcom:TEST:TYPE?*

## Test Information and Control

### Test Information

Time Elapsed

*:FETCh[1..n]:DATA:TELEcom:TEST:TIME?*

### Start/Stop/TX Button

**Start/Stop** button

*:SOURce[1..n]:DATA:TELEcom:TEST*  
*:SOURce[1..n]:DATA:TELEcom:TEST?*

**TX** button

*:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:STATus*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:STATus?*

### Reset Button

*:SOURce[1..n]:DATA:TELEcom:RESEt*

### Save/Laod Button

*:CONFig[1..n]:DATA:TELEcom:LOAD*  
*:CONFig[1..n]:DATA:TELEcom:SAVE*

### **Laser Button**

*:OUTPut[1..n]:TELEcom:LASer*  
*:OUTPut[1..n]:TELEcom:LASer?*

### **Discover Remote Button**

See *Discover Remote Button*.

## **Discover Remote Button**

### **Scan**

*:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SSUBnet*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SSUBnet?*

### **Target**

*:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SCANtarget:TYPE*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SCANtarget:TYPE?*

### **Status**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:REMote:RSCStatus?*

### **Table**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:DUALtest:STATistics?*

### **Loop Up**

*:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:LOOP:UP*

### **Loop Down**

*:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:LOOP:DOWN*

### **Connect**

*:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:CONNect*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:CONNect?*

### **Disconnect**

*:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:DISConnect*

## About (i) Button

### Module Details

**- Module ID**

*:FETCh[1..n]:DATA:TELEcom:MODUle:DETailS:CDATe?*

**- Serial Number**

*:FETCh[1..n]:DATA:TELEcom:MODUle:DETailS:SPVersion?*

Assembly Hardware Revision

*:SOURce[1..n]:DATA:TELEcom:VERDict:ENABLE*

**- Calibration Date**

*:SOURce[1..n]:DATA:TELEcom:VERDict:ENABLE?*

**- Software Product Verison**

*:SOURce[1..n]:DATA:TELEcom:REStore:DEFault*





# 5 *SCPI Command List - Setup*

## List of Pages

*BERT and Unframed BERT* on page 35

*CFP/CFP2* on page 38

*Clock* on page 39

*EtherBERT, and Unframed BERT* on page 41

*EtherSAM - Burst* on page 45

*EtherSAM - Global* on page 46

*EtherSAM - Ramp* on page 48

*FTFL/PT* on page 49

*Frequency* on page 51

*GFP-F/GFP-T* on page 52

*Interface (Ethernet)* on page 53

*Labels* on page 54

*MAC/IP/UDP* on page 55

*Modify Structure (Transport)* on page 60

*Modify Structure (Ethernet)* on page 62

*Network* on page 63

*ODU Channels - Global* on page 65

*RFC 2544 - Global* on page 67

*RFC 2544 - Subtests* on page 69

*RFC 6349* on page 72

*Services - Global* on page 74

*Services - Profile* on page 76

*Signal (Transport)* on page 79

*Signal - Signal Configuration (OTN)* on page 81

*Signal - Signal Configuration (SONET/SDH)* on page 82

*Smart Loopback* on page 83

## SCPI Command List - Setup

### *List of Pages*

---

*Streams - Global* on page 84

*Streams - Profile* on page 87

*System* on page 89

*Test Configurator* on page 90

*Timer* on page 91

*Traces (OTN)* on page 92

*Traces (SONET/SDH)* on page 96

## BERT and Unframed BERT

### Pattern

#### LINK

*:FETCh[1..n]:DATA:TELEcom:EOTN:ALARm:LINK?*

#### Coupled RX to TX

##### Framed

*:SENSe[1..n]:DATA:TELEcom:COUPled*

*:SENSe[1..n]:DATA:TELEcom:COUPled?*

##### Unframed

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:COUPled*

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:COUPled?*

#### No Pattern Analysis (Live)

*:SENSe[1..n]:DATA:TELEcom:PATtern:RXPanalysis:STATus*

*:SENSe[1..n]:DATA:TELEcom:PATtern:RXPanalysis:STATus?*

#### All Lanes

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL*

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL?*

#### TX Pattern

##### Framed

*:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE*

*:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE?*

*:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue*

*:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?*

##### Invert

*:SOURce[1..n]:DATA:TELEcom:POLarity*

*:SOURce[1..n]:DATA:TELEcom:POLarity?*

##### Unframed

*:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:TX*

*:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:TX?*

*:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX*

## SCPI Command List - Setup

### *BERT and Unframed BERT*

---

*:SOURCE[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX?*

#### **Invert**

*:SOURCE[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:TX*

*:SOURCE[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:TX?*

#### **RX Pattern**

##### **Framed**

*:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE*

*:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE?*

*:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue*

*:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?*

##### **Invert**

*:SENSe[1..n]:DATA:TELEcom:POLarity*

*:SENSe[1..n]:DATA:TELEcom:POLarity?*

##### **Unframed**

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX*

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX?*

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:RX*

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:RX?*

##### **Invert**

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:RX*

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:RX?*

#### **Pattern Sync**

*:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:SYNC?*

*:FETCh[1..n]:DATA:TELEcom:PATtern:GLOBal:ALARm:SYNC?*

#### **Bit Error**

##### **Pass/Fail Verdict and BER Threshold**

###### **- Framed**

*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:RATE*

*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:RATE?*

*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:COUNt*

*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:COUNt?*

*:SOURCE[1..n]:DATA:TELEcom:PATtern:VERDict:DISable*

- Unframed

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:RATE*  
*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:RATE?*  
*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:COUNt*  
*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:COUNt?*

## Service Disruption

### Disruption Monitoring

*:SENSe[1..n]:DATA:TELEcom:SDT*  
*:SENSe[1..n]:DATA:TELEcom:SDT?*

### Defect

*:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE*  
*:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE?*  
*:SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection*  
*:SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection?*

### No Defect time

*:SENSe[1..n]:DATA:TELEcom:SDT:NDTime*  
*:SENSe[1..n]:DATA:TELEcom:SDT:NDTime?*

### Pass/Fail Verdict

*:SENSe[1..n]:DATA:TELEcom:SDT:VERDict*  
*:SENSe[1..n]:DATA:TELEcom:SDT:VERDict?*

### SDT Threshold

*:SENSe[1..n]:DATA:TELEcom:SDT:THReshold*  
*:SENSe[1..n]:DATA:TELEcom:SDT:THReshold?*

## Restore OTN BERT Defaults

*:SOURce[1..n]:DATA:TELEcom:OTN:RESStore:DEFault*

## CFP/CFP2

### CFP/CFP2

*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:MODule:ID?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:VENDor:NAME?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:PART:NUMBer?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:SERial:NUMBer?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:REVisIon?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:FIRMware:VERSIon?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:CONNector:TYPE?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:SPEEd?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:TYPE?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:POWer:CLASs?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:LRATIo:TYPE?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:WDM:TYPE?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:CLEI:PRESEnce?*  
*:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:MODE?*

# Clock

## Clock Synchronization

### Clock Mode

*:INPut[1..n]:TELEcom:BACKplane:CLOCK*

*:INPut[1..n]:TELEcom:BACKplane:CLOCK?*

## Ext Clock In

### Interface

*:OUTPut[1..n]:TELEcom:LEVel*

*:OUTPut[1..n]:TELEcom:LEVel?*

*:OUTPut[1..n]:TELEcom:CLOCK:ALARm:STATus?*

### Termination Mode

*:OUTPut[1..n]:TELEcom:TERMination*

*:OUTPut[1..n]:TELEcom:TERMination?*

### Line Coding

*:OUTPut[1..n]:TELEcom:CODE*

*:OUTPut[1..n]:TELEcom:CODE?*

### Framing

*:OUTPut[1..n]:TELEcom:FRAMing*

*:OUTPut[1..n]:TELEcom:FRAMing?*

### Frequency

*:OUTPut[1..n]:TELEcom:CLOCK:FREQuency?*

### Offset

*:OUTPut[1..n]:TELEcom:CLOCK:FREQuency:OFFSet?*

## Ext Clock Out

### Interface

*:INPut[1..n]:TELEcom:LEVel*

*:INPut[1..n]:TELEcom:LEVel?*

## SCPI Command List - Setup

### Clock

---

#### **LBO**

*:INPut[1..n]:TELEcom:LBO*

*:INPut[1..n]:TELEcom:LBO?*

#### **Line Coding**

*:INPut[1..n]:TELEcom:CODE*

*:INPut[1..n]:TELEcom:CODE?*

#### **Framing**

*:INPut[1..n]:TELEcom:FRAMing*

*:INPut[1..n]:TELEcom:FRAMing?*

#### **Ref Out**

#### **Frequency**

*:INPut[1..n]:TELEcom:COUtput:FREQuency?*

*:INPut[1..n]:TELEcom:COUtput:STATUs?*

#### **Backplane**

#### **Backplane Clock**

*:INPut[1..n]:TELEcom:BCLock:ENABLE*

*:INPut[1..n]:TELEcom:BCLock:ENABLE?*

*:INPut[1..n]:TELEcom:CLOCK:ALARm:STATUs?*



## EtherBERT, and Unframed BERT

### Link

#### Link

*:FETCh[1..n]:DATA:TELEcom:EOTN:ALARm:LINK?*

### Pattern

#### Coupled RX to TX

##### Framed

*:SENSe[1..n]:DATA:TELEcom:COUPled*

*:SENSe[1..n]:DATA:TELEcom:COUPled?*

##### Unframed

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:COUPled*

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:COUPled?*

#### No Pattern Analysis (Live)

*:SENSe[1..n]:DATA:TELEcom:PATtern:RXPanalysis:STATus*

*:SENSe[1..n]:DATA:TELEcom:PATtern:RXPanalysis:STATus?*

#### All Lanes

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL*

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL?*

#### TX Pattern

##### Framed

*:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE*

*:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE?*

*:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue*

*:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?*

##### Invert

*:SOURce[1..n]:DATA:TELEcom:POLarity*

*:SOURce[1..n]:DATA:TELEcom:POLarity?*

## SCPI Command List - Setup

*EtherBERT, and Unframed BERT*

---

### Unframed

*:SOURCE[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:TX*  
*:SOURCE[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:TX?*  
*:SOURCE[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX*  
*:SOURCE[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX?*

### Invert

*:SOURCE[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:TX*  
*:SOURCE[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:TX?*

### RX Pattern

#### Framed

*:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE*  
*:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE?*  
*:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue*  
*:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?*

#### Invert

*:SENSe[1..n]:DATA:TELEcom:POLarity*  
*:SENSe[1..n]:DATA:TELEcom:POLarity?*

#### Unframed

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX*  
*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX?*  
*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:RX*  
*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:RX?*

#### Invert

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:RX*  
*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:RX?*

### Pattern Sync

*:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:SYNC*  
*:FETCh[1..n]:DATA:TELEcom:PATtern:GLOBal:ALARm:SYNC?*

### Bit Error

#### Pass/Fail Verdict and BER Threshold

*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:RATE*  
*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:RATE?*

*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:COUNT*  
*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:COUNT?*  
*:SOURce[1..n]:DATA:TELEcom:PATtern:VERDict:DISable*

**Unframed**

*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:RATE*  
*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:RATE?*  
*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:COUNT*  
*:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:COUNT?*  
*:SOURce[1..n]:DATA:TELEcom:PATtern:VERDict:DISable*

## Service Disruption

### Disruption Monitoring

*:SENSe[1..n]:DATA:TELEcom:SDT*  
*:SENSe[1..n]:DATA:TELEcom:SDT?*

### No Defect Time

*:SENSe[1..n]:DATA:TELEcom:SDT:NTTime*  
*:SENSe[1..n]:DATA:TELEcom:SDT:NTTime?*

### Pass/Fail Verdict

*:SENSe[1..n]:DATA:TELEcom:SDT:VERDict*  
*:SENSe[1..n]:DATA:TELEcom:SDT:VERDict?*

### SDT Threshold

*:SENSe[1..n]:DATA:TELEcom:SDT:THReshold*  
*:SENSe[1..n]:DATA:TELEcom:SDT:THReshold?*

## Shapping

### TX Rate

*:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RATE*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RATE?*

### Enable TX

*:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TX:STATus*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TX:STATus?*

## Ethernet Frame

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE?*

## EtherSAM - Burst

### Parameters

#### Number of Burst Sequence

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSe  
quence*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSe  
quence?*

#### Refil Delay Ratio

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERat  
io*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERat  
io?*

#### Burst/IR Frame Ratio

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BiRFra  
me*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BiRFra  
me?*

### Table

#### CBS Test Time

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:CBSt:TIME?*

#### EBS Test Time

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:EBSt:TIME?*

#### Total Burst Test Time

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:TBURSt:TIME?*

#### Total

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:TOTAl?*

## EtherSAM - Global

### Dual Test Set

#### Dual Test Set

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:DUALtest:ENABLEd*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:DUALtest:ENABLEd?*

#### Discover Remote Button

Refer to *Discover Remote Button*.

### Subtests

#### Service Configuration Test

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE?*

#### Service Performance Test

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTest:ENABLEd*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTest:ENABLEd?*

#### Subtest Duration

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTest:DURATION*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTest:DURATION?*

#### Global Test Duration Estimation

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:TDURATION:ESTimate?*

### Global Options

#### Per Direction Configuration

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:PDIREction:CONFig:STATUs*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:PDIREction:CONFig:STATUs*  
*?*

#### Latency Measurement Mode

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:LMMode*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:LMMode?*

**LOPPS-L and LOPPS-R**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:LATency:ALARm:CURRent?*

**Restore EtherSAM Defaults**

*:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RESTore:DEFault*

## EtherSAM - Ramp

### **Add Step**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:ADD*

### **Delete Step**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DELEte*

### **Defaults**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DEFault*

### **Step Time**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME?*

### **Ramp Duration**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:DURation?*



## FTFL/PT

### FTFL

#### **Fault Indication**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication?*

#### **Code**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE?*

#### **Operator Identifier**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENtifier*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENtifier?*

#### **Operator Specific**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPec*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPec?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:OPSPec*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:OPSPec?*

### PT / Global PT

#### **Payload Type**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?*

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe*

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?*

## SCPI Command List - Setup

*FTFL/PT*

---

*:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe*  
*:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe?*

### **Payload Type - Global Overwrite Status Icon (Multi-Channel OTN)**

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:GOVEverwrite?*

### **Code**

*:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE*  
*:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?*  
  
*:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE*  
*:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE:RECEived?*

### **OPU-PLM**

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PLM*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PLM?*  
  
*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PLM*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PLM?*

## Frequency

### TX Frequency

#### Frequency

*:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency?*

#### Offset

*:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet*

*:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet?*

*:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue*

*:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue?*

### RX Frequency

#### Frequency

*:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency?*

#### Offset

*:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue?*

#### Max Negative Offset

*:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:NEGative?*

#### Max Positive Offset

*:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:POSitive?*

## GFP-F/GFP-T

### **CDF pFCS / CMF pFCS**

*:SOURCE[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:TYPE?*

### **EXI**

*:SOURCE[1..n]:DATA:TELEcom:GFP:CONFig:EXI*

*:SOURCE[1..n]:DATA:TELEcom:GFP:CONFig:EXI?*

### **CID**

*:SOURCE[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:CID*

*:SOURCE[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:CID?*

## Interface (Ethernet)

### LINK

#### Physical Interface

##### **TX Power**

*:SENSe[1..n]:DATA:TELEcom:OPTical:TX:POWer?*

*:SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer?*

##### **Min RX Power**

*:SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer:MINimum?*

*:SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer:MAXimum?*

##### **Wavelength**

*:SENSe[1..n]:DATA:TELEcom:OPTical:WAVelength?*

##### **Laser ON/OFF button**

Refer to *Laser ON/OFF Button*

##### **Power Range**

*:SENSe[1..n]:DATA:TELEcom:OPTical:POWer:RANGe?*

##### **Laser OFF at Start-Up**

*:SENSe[1..n]:DATA:TELEcom:LOFF*

*:SENSe[1..n]:DATA:TELEcom:LOFF?*

## Labels

### **STS/AU Path (C2)**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel*

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel?*

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel:EXPEcted?*

### **PLM-P/UNEQ-P / HP-PLM/HP-UNEQ**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PUNeq*

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PUNeq?*

## MAC/IP/UDP

### MAC

#### Source MAC Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:SOURce?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:SOURce:IP*

#### Destination MAC Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination?*

#### EtherType

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ETHer?*

#### Source Flooding

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:SOURce:FLOoding*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:SOURce:FLOoding?*

#### Destination Flooding

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination:FLOoding*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination:FLOoding*

*?*

#### Flood Range

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:FLOoding:RANGe*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:FLOoding:RANGe?*

### VLAN

#### VLAN ID

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID?*

## SCPI Command List - Setup

MAC/IP/UDP

---

### Priority

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:PRiority*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:PRiority?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRiority*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRiority?*

### Type

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE?*

### Drop Eligible

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID:ELIGiblebit*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID:ELIGiblebit?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGible  
bit*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGible  
bit?*

## MPLS

### Label

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:LABel*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:LABel?*

### COS

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSexp*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSexp?*

### TTL

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL?*



## IP (IPv4)

### Automatic IP (DHCP)

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:IP:AUTomatic:STATus*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:IP:AUTomatic:STATus?*

### Source IP Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:SOURce:IP*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:SOURce:IP?*

### Destination IP Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination:IP*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination:IP?*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:QPING*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:QPING?*

### Subnet Mask

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:MASK:IP*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:MASK:IP?*

### Source IP Multiplier

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:MULTiplicat*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:MULTiplicat?*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:RANGe*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:RANGe?*

### Resolve MAC Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve?*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve:STATus?*

### Default Gateway

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway?*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDReSS*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDReSS?*

### **TTL**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL?*

### **IP TOS/DS**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs?*

## **IPv6**

### **IPv6 Destination IP Address**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IPVersion?*

### **Resolve MAC Address**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve?*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve:STATus?*

### **Source IP Multiplier**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:MULTiplicat*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:MULTiplicat?*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:RANGE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:RANGE?*

### **HOP Limit**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:IPVersion:HOP:LIMit*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:IPVersion:HOP:LIMit?*

### **Flow Control**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FLABel:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FLABel:IPVersion?*

### **TOS/DS (Traffic Class)**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs?*

## UDP

### Source Port

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT?*

### Destination Port

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT?*

## Payload

### Pattern

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PAYLoad*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PAYLoad?*

## Modify Structure (Transport)

### Interface/Rate

*:SOURCE[1..n]:DATA:TELEcom:ITYPE*

*:SOURCE[1..n]:DATA:TELEcom:ITYPE?*

### Connector

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver?*

### Framing

OTN:

*:SOURCE[1..n]:DATA:TELEcom:OTN:FRAMing*

*:SOURCE[1..n]:DATA:TELEcom:OTN:FRAMing?*

SONET/SDH

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:FRAMing?*

### OTN Multiplexing

*:SOURCE[1..n]:DATA:TELEcom:ODU:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:ODU:TYPE?*

### Embedded SONET/SDH

*:SOURCE[1..n]:DATA:TELEcom:OTN:MULTiplex:ITYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:MULTiplex:ITYPE?*

### SONET/SDH Multiplexing

*:SOURCE[1..n]:DATA:TELEcom:SONet:TEST:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:SONet:TEST:TYPE?*

### Client

OTN

*:SOURCE[1..n]:DATA:TELEcom:OTN:CLient*

*:SOURCE[1..n]:DATA:TELEcom:OTN:CLient?*

SONET/SDH

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:CLient?*

**Topology**

*:SOURCE[1..n]:DATA:TELEcom:TOPology*

*:SOURCE[1..n]:DATA:TELEcom:TOPology?*

## Modify Structure (Ethernet)

### **Interface**

*:SOURCE[1..n]:DATA:TELEcom:ITYPE*

*:SOURCE[1..n]:DATA:TELEcom:ITYPE?*

### **Connector**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver?*

### **Framing**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:BERT:FRAMing*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:BERT:FRAMing?*

### **Topology**

*:SOURCE[1..n]:DATA:TELEcom:TOPology*

*:SOURCE[1..n]:DATA:TELEcom:TOPology?*

## Network

### MAC

#### MAC Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:MAC:ADDRESS*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:MAC:ADDRESS?*

#### Factory Default

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault?*

#### Frame Format

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DATAlink:TYPE?*

### IP

#### IP Version

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion?*

#### IP Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:ADDRESS:IP*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:ADDRESS:IP?*

#### Link-Local IPv6 Address

*1000056\_Ip\_Ipv6\_Lla\_Address*  
*1000056\_Ip\_Ipv6\_Lla\_Address?*

#### Global IPv6 Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:ADDRESS*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:ADDRESS?*

#### Automatic IP (DHCP)

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DHCP:STATUS*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DHCP:STATUS?*

#### **Subnet Mask**

*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:SUBNet:MASK*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:SUBNet:MASK?*

#### **Default Gateway**

*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:DEFault:GATeway:STATus*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:DEFault:GATeway:STATus?*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:DEFault:GATeway:ADDResS*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:DEFault:GATeway:ADDResS?*

## **VLAN**

#### **VLAN Tag**

*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN?*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN:STACkEd*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN:STACkEd?*

#### **VLAN ID**

*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN:ID*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN:ID?*

#### **Type**

*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN:TYPE*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN:TYPE?*

#### **Priority**

*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN:PRIority*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN:PRIority?*

#### **Drop Eligible**

*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN:ID:ELIGiblebit*  
*:SOURCE[1..n]:DATA:TELECOM:ETHernet:NETWork:VLAN:ID:ELIGiblebit?*



## ODU Channels - Global

### Pattern

#### No Pattern Analysis (Live)

*:SENSe[1..n]:DATA:TELEcom:PATtern:RXPanalysis:STATus*

*:SENSe[1..n]:DATA:TELEcom:PATtern:RXPanalysis:STATus?*

#### TX Pattern

*:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE*

*:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE?*

#### Invert

*:SOURce[1..n]:DATA:TELEcom:POLarity*

*:SOURce[1..n]:DATA:TELEcom:POLarity?*

### Bit Error

#### Pass/Fail Verdict and BER Threshold

*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:RATE*

*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:RATE?*

*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:COUNT*

*:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:COUNT?*

*:SOURce[1..n]:DATA:TELEcom:PATtern:VERDict:DISable*

### Service Disruption

#### Disruption Monitoring

*:SENSe[1..n]:DATA:TELEcom:SDT*

*:SENSe[1..n]:DATA:TELEcom:SDT?*

#### Defect

*:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE*

*:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE?*

*:SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection*

*:SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection?*

## SCPI Command List - Setup

*ODU Channels - Global*

---

### **No Defect time**

*:SENSe[1..n]:DATA:TELEcom:SDT:NTTime*

*:SENSe[1..n]:DATA:TELEcom:SDT:NTTime?*

### **Pass/Fail Verdict**

*:SENSe[1..n]:DATA:TELEcom:SDT:VERDict*

*:SENSe[1..n]:DATA:TELEcom:SDT:VERDict?*

### **SDT Threshold**

*:SENSe[1..n]:DATA:TELEcom:SDT:THReshold*

*:SENSe[1..n]:DATA:TELEcom:SDT:THReshold?*

## RFC 2544 - Global

### Dual Test Set

#### Dual Test Set

`:SOURCE[1..n]:DATA:TELEcom:ETHernet:DUALtest:ENABLEd`

`:SOURCE[1..n]:DATA:TELEcom:ETHernet:DUALtest:ENABLEd?`

#### Discover Remote Button

Refer to *Discover Remote Button*.

### Global Options

#### Flow Direction

`:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FDIREction`

`:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FDIREction?`

#### Pass/Fail Verdict

`:FETCH[1..n]:DATA:TELEcom:MODule:DEtails:MID?`

`:FETCH[1..n]:DATA:TELEcom:MODule:DEtails:SNUMber?`

### Subtests

#### Throughput

`:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:ENABLE`

`:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:ENABLE?`

`:FETCH[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:MINTime?`

#### Back-to-Back

`:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ENABLE`

`:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ENABLE?`

`:FETCH[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MINTime?`

#### Frame Loss

`:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:ENABLE`

`:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:ENABLE?`

`:FETCH[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:MINTime?`

## SCPI Command List - Setup

*RFC 2544 - Global*

---

### **Latency**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ENABLE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ENABLE?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MINTime?*

### **Total**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TOTal:MINTime?*

## **Frame Distribution**

### **Frame Distribution**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution?*

### **Quantity**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity?*

### **Frame Size**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZE?*

## **Restore RFC 2544 Defaults**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:RESTore*

## RFC 2544 - Subtests

### Throughput

#### Max. Rate

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:MAXRate*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:MAXRate?*

#### Trial Duration

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:TTIME*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:TTIME?*

#### Trials

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:TAVerage*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:TAVerage?*

#### Accuracy

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:ACCuracy*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:ACCuracy?*

#### Acceptable Errors

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:AERRors*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:AERRors?*

#### Validations

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:VALidations*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:VALidations?*

### Back-to\_Back

#### Burst Time

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames?*

#### Trials

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage?*

## SCPI Command List - Setup

### RFC 2544 - Subtests

---

#### **Accuracy**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy?*

#### **Acceptable Errors**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors?*

#### **Bursts**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:NBURst?*

## Frame Loss

#### **Max. Rate**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:MAXRate*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:MAXRate?*

#### **Trial Duration**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TTIME*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TTIME?*

#### **Trials**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TAVerage*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TAVerage?*

#### **Granularity**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TGRanularity*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:TGRanularity?*

## Latency

#### **Margin**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MARGin*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MARGin?*

#### **Trial Duration**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TTIME*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TTIME?*

**Trials**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TAVerage*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TAVerage?*

**Measurement Mode**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:LMMode*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:LMMode?*

**Copy From Throughput**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPIYtest*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPIYtest?*

## RFC 6349

### Connection

#### Operation Mode

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE?*

#### Direction

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:DIRection*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:DIRection?*

### Parameters

#### Multiple Connections

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:MULTiple:CONNECTIONs*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:MULTiple:CONNECTIONs?*

#### CIR

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:CIR*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:CIR?*

#### TOS/DS

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TOSDs*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TOSDs?*

#### TCP Server Port

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort?*

### MTU

#### Max MTU

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:MAX:MTU*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:MAX:MTU?*

#### Path MTU Discovery

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:PATH:MTU:DISCcovery*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:PATH:MTU:DISCcovery?*



## Window Sweep

### Window Sweep

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep?*

### Duration (per step)

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:DURation*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:DURation?*

## TCP Throughput

### Duration

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:DURation*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:DURation?*

### Pass/Fail Verdict

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:VERDict*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:VERDict?*

### Threshold (% of ideal)

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold?*

## Restore RFC 6349 Defaults

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:REStore:DEFault*

## Services - Global

### Service Name

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NAME*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NAME?*

## Addressing - Batch - Addressing Batch Config

### Source IP Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABLE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABLE?*

### Couple with Interface / Automatic IP (CHCP) / Set to

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDResS:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDResS:TYPE?*

### IP Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP?*

### Subnet Mask

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:SUBNet:MASK*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:SUBNet:MASK?*

### Default Gateway

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:ENABLE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:ENABLE*

*?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:IP*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:IP?*

### Destination MAC Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ENABLE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ENABLE*

*?*

### Resolve / Set to

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:TYPE?*

**MAC Address**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ADDRe  
ss*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ADDRe  
ss?*

**IP Address****Destination IP Address**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP:ENABle*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP:ENABle?*

**IP Address**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP?*

**(UN)Select All / Invert**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy*

**Apply To**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:STReam*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:STReam?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:APPLy*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:SYNC:PROGress?*

## Services - Profile

### Service

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle?*

### Profile

#### Profile button

Refer to *Profile (Services)*.

#### EMIX - Quantity

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity?*

#### EMIX Frame Sizes

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMesize*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMesize?*

#### EMIX - Restore Default

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:RESDefaul*  
*t*

### Test Parameters

#### Traffic Policing

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:TRAPolicing:ENABle*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:TRAPolicing:ENABle?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:TRAPolicing:VALue*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:TRAPolicing:VALue?*

### **Burst Max Rate**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:BMRate*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:BMRate?*

## **SLA Parameters**

### **Information Rate**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:INFRate:ENABLE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:INFRate:ENABLE?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:INFRate*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:INFRate?*

### **Performance Criteria**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:PERCriteria:ENABLE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:PERCriteria:ENABLE?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:PERCriteria:VALue*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:PERCriteria:VALue?*

### **Burst Size**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:BSIZE:ENABLE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPARAmeter:BSIZE*

## SCPI Command List - Setup

### *Services - Profile*

---

*IZe:ENABle?*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BS*

*IZe*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BS*

*IZe?*

## Signal (Transport)

### Physical Interface

#### **TX Power**

*:SENSe[1..n]:DATA:TELEcom:OPTical:TX:POWer?*

*:SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer?*

#### **Min RX Power**

*:SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer:MINimum?*

*:SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer:MAXimum?*

#### **Wavelength**

*:SENSe[1..n]:DATA:TELEcom:OPTical:WAVelength?*

#### **Power Range**

*:SENSe[1..n]:DATA:TELEcom:OPTical:POWer:RANGe?*

#### **Laser ON/OFF button**

Refer to *Laser ON/OFF Button*

#### **Laser OFF at Start-Up**

*:SENSe[1..n]:DATA:TELEcom:LOFF*

*:SENSe[1..n]:DATA:TELEcom:LOFF?*

### TX Frequency

#### **Frequency**

*:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQUency?*

#### **Offset**

*:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQUency:OFFSet*

*:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQUency:OFFSet?*

*:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQUency:OFFSet:VALue*

*:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQUency:OFFSet:VALue?*

### **RX Frequency**

#### **Frequency**

*:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQUency?*

#### **Max Offset**

*:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQUency:NEGative?*

*:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQUency:POSitive?*

#### **Offset**

*:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQUency:OFFSet:VALue?*



## Signal - Signal Configuration (OTN)

### OTU

#### FEC

*:SOURCE[1..n]:DATA:TELEcom:OTN:FEC*

*:SOURCE[1..n]:DATA:TELEcom:OTN:FEC?*

#### Scrambler

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler?*

#### Background Traffic

*:SOURCE[1..n]:DATA:TELEcom:OTN:BTRaffic:PT[1..n]*

*:SOURCE[1..n]:DATA:TELEcom:OTN:BTRaffic:PT[1..n]?*

## Signal - Signal Configuration (SONET/SDH)

### OC/STM

#### Synchronization Status Message (S1)

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:S:BITS:SSMessage*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:S:BITS:SSMessage?*

#### REI-L Computation Method

*:SOURCE[1..n]:DATA:TELEcom:BACKground:COMPutation*  
*:SOURCE[1..n]:DATA:TELEcom:BACKground:COMPutation?*

### STS/AU and VT/TU Mappings

#### Timeslot/Number

*:SOURCE[1..n]:DATA:TELEcom:POSition*  
*:SOURCE[1..n]:DATA:TELEcom:POSition?*

#### TCM

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABle*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABle?*  
*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABle*  
*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABle?*

#### TC-UNEQ-P / HPTC-UNEQ

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TCUNeq:ENABle*  
*:SENSe[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TCUNeq:ENABle?*

#### Overwrite Fixed Stuff

*:SOURCE[1..n]:DATA:TELEcom:BACKground:BULK*  
*:SOURCE[1..n]:DATA:TELEcom:BACKground:BULK?*

#### Background

*:SOURCE[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP*  
*:SOURCE[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP?*

## Smart Loopback

### Loopback

#### Mode

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:SLOopback:MODE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:SLOopback:MODE?*

### Matching & Swapping

#### MAC Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:SLOopback:MATCHing:MAC:ADDRESS:MODE?*

#### IP Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:ADDRESS:IP?*

#### IP LLA

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWORK:LOCAL:IPVersion:ADDRESS?*

#### IP GUA

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWORK:GLOBAL:IPVersion:ADDRESS?*

#### UDP Port

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:SLOopback:MATCHing:UDP:PORT:MODE?*

## Streams - Global

### Enable

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled?*

### Stream Name

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NAME*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NAME?*

## Addressing - Batch - Addressing Batch Config

### Source IP Address

#### Source IP Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABle*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABle?*

#### Couple with Interface / Automatic IP (CHCP) / Set to

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDRes:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDRes:TYPE?*

#### IP Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP?*

#### Subnet Mask

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:SUBNet:MASK*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:SUBNet:MASK?*

#### Default Gateway

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:ENABle*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:ENABle*  
*?*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:IP*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:IP?*

## Destination MAC Address

### Destination MAC Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ENABLE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ENABLE*  
*?*

### Resolve / Set to

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:TYPE?*

### MAC Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ADDRe*  
*SS*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ADDRe*  
*SS?*

## Destination IP Address

### Destination IP Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP:ENABLE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP:ENABLE?*

### IP Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP?*

## (UN)Select All / Invert

### (UN)Select All / Invert

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPI*

### **Apply To**

#### **Apply To**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPI:STReam*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPI:STReam?*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPI:APPLY*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPI:SYNC:PROGress?*

### **Global Options**

#### **QoS Metrics Tag Insertion**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABle*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABle?*

### **Restore Default**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:REStore:DEFault*

## Streams - Profile

### Stream

#### Stream Name

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NAME*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NAME?*

#### Enable

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled?*

### Profile

#### Profile

Refer to *Profile (Stream)*.

#### Frame Size

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE:TYPE?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE?*

#### Sweep

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:STARt*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:STARt?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:END*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:END?*

### Shaping

#### TX Mode

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:MODE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:MODE?*

#### TX Rate / Max TX Rate

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:RATE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:RATE?*

## SCPI Command List - Setup

### *Streams - Profile*

---

#### **Frame Count**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame  
:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame?*

#### **Total TX Rate**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ENABled:BANDwidth?*

#### **Link Capacity**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:TOTAL:BANDwidth?*



## System

### Factory Default

#### Restore Default

*:SOURce[1..n]:DATA:TELEcom:FACTory:RESTore:DEFault*

## Test Configurator

Transceiver Status

*:FETCh[1..n]:DATA:TELEcom:OPTical:MODule:STATus?*

## Timer

### Timer

**Duration / Start Time / Stop Time** check boxes

*:SOURCE[1..n]:DATA:TELEcom:TIMer:CONFig*

*:SOURCE[1..n]:DATA:TELEcom:TIMer:CONFig?*

### Duration

*:SOURCE[1..n]:DATA:TELEcom:TIMer:DURation*

*:SOURCE[1..n]:DATA:TELEcom:TIMer:DURation?*

*:SOURCE[1..n]:DATA:TELEcom:TIMer:UDEF*

*:SOURCE[1..n]:DATA:TELEcom:TIMer:UDEF?*

### Start Time

*:SOURCE[1..n]:DATA:TELEcom:TIMer:STARt*

*:SOURCE[1..n]:DATA:TELEcom:TIMer:STARt?*

### Stop Time

*:SOURCE[1..n]:DATA:TELEcom:TIMer:STOP*

*:SOURCE[1..n]:DATA:TELEcom:TIMer:STOP?*

### ARM

*:SOURCE[1..n]:DATA:TELEcom:TIMer*

*:SOURCE[1..n]:DATA:TELEcom:TIMer?*

## Traces (OTN)

### OTU - SM TTI Traces

#### SAPI

##### - Received Message

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B?*

##### - Expected Message

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:EXPeCted*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:EXPeCted?*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPeCted*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPeCted?*

#### DAPI

##### - Received Message

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B?*

##### - Expected Message

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:EXPeCted*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:EXPeCted?*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPeCted*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPeCted?*

#### Operator Specific

##### - Received Message

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B32*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B32?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:OPSPec:B*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:OPSPec:B?*

### **SAPI/DAPI OTU-TIM**

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM?*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM?*

### **ODU - PM TTI Traces**

Channel selection for Multi-Channel OTN

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHannel*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHannel?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHANnel*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHANnel?*

**Expected Message (SAPI and DAPI)** for Multi-Channel OTN: Icon indicating that at least one channel uses a different message.

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:GOVEverwrite?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:GOVEverwrite?*

### **SAPI**

#### **- Received Message**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:SAPI:B*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:SAPI:B?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:SAPI:B*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:SAPI:B?*

#### **- Expected Message**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPeCted*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPeCted?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:EXPeCted*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:EXPeCted?*

## SCPI Command List - Setup

### Traces (OTN)

---

#### **DAPI**

##### **- Received Message**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:DAPI:B*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:DAPI:B?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:DAPI:B*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:DAPI:B?*

##### **- Expected Message**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXPeCted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXPeCted?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:EXPeCted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:EXPeCted?*

#### **Operator Specific**

##### **- Received Message**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:OPSPec:B*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:OPSPec:B?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:OPSPec:B*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:OPSPec:B?*

#### **SAPI/DAPI ODU-TIM**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:TIM*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:TIM?*

## **ODU - TCM TTI Traces**

#### **SAPI**

##### **- Received Message**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:SAPI:B*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:SAPI:B?*

**- Expected Message**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted?*

**DAPI****- Received Message**

*:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B*

*:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B?*

*:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:DAPI:B*

*:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:DAPI:B?*

**- Expected Message**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPEcted?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:EXPEcted?*

**Operator Specific****- Received Message**

*:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:OPSPec:B*

*:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:OPSPec:B?*

*:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:OPSPec:B*

*:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:OPSPec:B?*

**SAPI/DAPI TCM-TIM**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:TIM*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:TIM?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:TIM*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:TIM?*

## Traces (SONET/SDH)

### Traces

#### Section/RS (J0)

**- Format**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATtern*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATtern?*

**- Generated**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATtern:B*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATtern:B?*

#### STS/AU/TU-3 Path (J1)

**- Format**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATtern*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATtern?*

**- Generated**

**- Format**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATtern:B*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATtern:B?*

#### TIM-S/RS-TIM

**- Enable**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM*  
*:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM?*

**- Format**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern*  
*:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern?*

**- Expected**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern:B*  
*:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern:B?*



**TIM-P/HP-TIM**

- Enable

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM*

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM?*

- Format

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern*

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern?*

- Expected

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern:B*

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern:B?*

**TCM Access Point Identifier****STS/AU Path (N1)**

- Generated

*:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:MESSAge*

*:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:MESSAge?*

- Expected

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:EXpected*

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:EXpected?*

- Enable

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:TCTim*

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:TCTim?*



## List of Pages

*Alarms/Errors* on page 100

*FTFL/PT* on page 134

*GFP-F/GFP-T* on page 136

*Labels* on page 137

*Logger and Alarms/Errors Logger* on page 138

*MPLS* on page 139

*OTL-SDT* on page 140

*Performance Monitoring* on page 141

*SDT (Multi-Channel OTN)* on page 142

*Service Configuration - Burst* on page 144

*Service Configuration - Ramp* on page 145

*Service Performance* on page 146

*Streams - Frame Loss / Out-of-Sequence* on page 148

*Streams - Jitter* on page 149

*Streams - Latency* on page 150

*Streams - Throughput* on page 151. *Summary* on page 152

*Summary* on page 152

*Summary (EtherSAM)* on page 155

*Summary (Multi-Channel OTN)* on page 157

*Summary (RFC 2544)* on page 158

*Summary (RFC 6349)* on page 160

*Summary (Traffic Gen & Mon)* on page 162

*Traces - OTN* on page 163

*Traces - SONET/SDH* on page 167

*Traffic - Ethernet* on page 169

*Traffic - Flow Control* on page 170

## Alarms/Errors

*BER | CLOCK | Ethernet | Ethernet - PCS Lanes | FEC | GFP | GMP | Interface | IP/UDP | ODUx | ODUx-TCM | OPUx | OTL | OTUx | QoS Metrics | Section/Line/RS/MS | STS-x / AU-x | TCM (SONET/SDH) | Transcoding |*

### Global Layer

*:FETCh[1..n]:DATA:TELEcom:ALARm:HISTory?  
:FETCh[1..n]:DATA:TELEcom:ALARm:SEConds?  
:FETCh[1..n]:DATA:TELEcom:ALARm:CURRent?  
:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?  
:FETCh[1..n]:DATA:TELEcom:ERRor:SEConds?  
:FETCh[1..n]:DATA:TELEcom:ERRor:CURRent?  
:FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?  
:FETCh[1..n]:DATA:TELEcom:ERRor:RATE?  
:FETCh[1..n]:DATA:TELEcom:ERRor:COUNT:TOTal?  
:FETCh[1..n]:DATA:TELEcom:ERRor:RATE:TOTal?*

### BER

#### Alarms

*:FETCh[1..n]:DATA:TELEcom:PATTern:ALARm:PATTern:SEConds?  
:FETCh[1..n]:DATA:TELEcom:PATTern:ALARm:PATTern:HISTory?  
:FETCh[1..n]:DATA:TELEcom:PATTern:ALARm:PATTern:CURRent?*

#### Errors

*:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:PATTern:SEConds?  
:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:PATTern:COUNT?  
:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:PATTern:RATE?  
:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:PATTern:HISTory?  
:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:PATTern:CURRent?*

#### Injection

**- Channel**

*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CHAN*  
*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CHAN?*  
*:SOURCE[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CHAN*  
*:SOURCE[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CHAN?*

**Mode Manual****- Defect**

*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE?*

**- Amount**

*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AMount*  
*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AMount?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:INJect*

**Mode Rate****- Defect**

*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE?*

**- Rate**

*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:RATE*  
*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:RATE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:CONTinuous*  
*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:CONTinuous?*

**Mode Max Rate****- Defect**

*:SOURCE[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:TYPE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated*  
*:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated?*

**CLOCK****Alarms**

## SCPI Command List - Results

### Alarms/Errors

---

*:INPut[1..n]:TELEcom:BACKplane:ALARm:STATus:HISTory?*  
*:INPut[1..n]:TELEcom:BACKplane:ALARm:STATus:CURRent?*  
*:INPut[1..n]:TELEcom:BACKplane:ALARm:STATus:SECOnds?*

## Ethernet

### Alarms

- Up to 10GE

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:GLOBal:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:GLOBal:SECOnds?*

- 40GE/100GE including global PCS Lanes

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:LINK?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:SECOnds?*

- Ethernet over OTN:

*:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:SECOnds?*

### Errors

- Up to 10GE

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:RATE?*

- 40GE/100GE

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:RATE?*

**Oversize Monitoring**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:OVERsize*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:OVERsize?*

**Injection****Mode Manual****- Defect**

Up to 10GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:MANual:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:MANual:TYPE?*  
40GE/100GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE?*

**- Amount**

Up to 10GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AMOUNT*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AMOUNT?*  
40GE/100GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUNT*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUNT?*

**- Inject**

Up to 10GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:INJECT*  
40GE/100GE  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJECT*

**Mode Rate****- Defect**

Up to 10GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:TYPE?*  
40GE/100GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:TYPE?*

**- Rate**

Up to 10GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:RATE*

## SCPI Command List - Results

### Alarms/Errors

---

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:RATE?  
40GE/100GE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE  
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE?*

#### **- Inject**

Up to 10GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated  
40GE/100GE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated?*

#### **Mode Max Rate**

##### **- Inject**

Up to 10GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:CONTInuous*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:CONTInuous?  
40GE/100GE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTInuous*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTInuous?*

#### **Mode Continuous**

##### **- Defect**

Up to 10GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:TYPE 40GE/100GE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE?*

##### **- Inject**

Up to 10GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe?*

40GE/100GE

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical*

## Ethernet - PCS Lanes

### Alarms



*:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:CURREnt?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:SECOnds?*

**Errors**

- Global

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:GLOBal:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:GLOBal:CURREnt?*

- Per Lane

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:CURREnt?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:RATE?*

- **Total**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:COUNt:TOTal?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:RATE:TOTal?*

**Skew Alarm Threshold**

*:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THREshold*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THREshold?*

- **Default** button

*:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THREshold:DEFault*

**Injection**

**Lane selection**

*:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:LANE*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:LANE?*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:ALANes*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:ALANes?*

**Mode Manual**

- **Defect**

*:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE*  
*:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE?*

- **Amount**

## SCPI Command List - Results

### Alarms/Errors

---

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:INJECT*

#### **Mode Rate**

**- Defect**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:TYPE?*

**- Rate**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:RATE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:RATE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated?*

#### **Mode Max Rate**

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTinuous*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTinuous*

*?*

## **FEC**

### **Errors**

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:SECOnds?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:CURREnt?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:COUNt?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:RATE?*

### **Injection**

#### **Mode Manual**

**- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:MANual:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:MANual:TYPE?*

**- Amount**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AMOUnt*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AMOUnt?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:INJect*

**Mode Rate****- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:TYPE?*

**- Rate**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:RATE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:RATE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated?*

**Mode Max Rate****- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:CONTinuous*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:CONTinuous?*

**GFP****Alarms**

*:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:SECOnds?*

*:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:SECOnds?*

**Errors**

## SCPI Command List - Results

### Alarms/Errors

---

*:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:RATE?*

*:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:RATE?*

### Reserved CMF Monitoring

*:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:CMF*  
*:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:CMF?*

### Injection

#### Mode Manual

##### - Defect

*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE*  
*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE?*  
*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:MANual:TYPE*  
*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:MANual:TYPE?*

##### - Amount

*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AMOUNt*  
*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AMOUNt?*  
*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AMOUNt*  
*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AMOUNt?*

##### - Inject

*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:INJect*  
*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:INJect*

#### Mode Rate

##### - Defect

*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTOmated:TYPE*  
*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTOmated:TYPE?*  
*:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTOmated:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:CHANnel:AUTomated:TYPE?*

**- Rate**

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:FRAME:AUTomated:RATE*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:FRAME:AUTomated:RATE?*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:CHANnel:AUTomated:RATE*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:CHANnel:AUTomated:RATE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:FRAME:AUTomated*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:FRAME:AUTomated?*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:CHANnel:AUTomated*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:CHANnel:AUTomated?*

**Mode Max Rate**

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:FRAME:AUTomated:CONTinuous*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:FRAME:AUTomated:CONTinuous?*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:CHANnel:AUTomated:CONTinuous*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:CHANnel:AUTomated:CONTinuous?*

**Mode Continuous**

**- Defect**

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:FRAME:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:FRAME:TYPE?*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:TYPE?*

**- Period**

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:PERiod*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:PERiod?*

**- User-Defined UPI**

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:FRAME*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:FRAME?*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel*

*:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel?*

## SCPI Command List - Results

### Alarms/Errors

---

## GMP

### Alarms

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:SEConds?*

### Errors

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:COUnT?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:RATE?*

## Interface

### Alarms

#### - LOS (40/100G)

*:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:SEConds?*

#### - 40GE/100GE, Serial OTU3/OTU3e1/OTU3e2, OC-768/STM256

*:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:TX:STATus?*

#### - Optical

*:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:RX:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:RX:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:RX:SEConds?*

- Per Lane

*:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:SECOnds?*

## **Injection**

### **Mode Continuous**

- Lane

*:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE*

*:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE?*

*:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:ALANes*

*:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:ALANes?*

- Defect

*:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:TYPE?*

- Inject

*:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT*

*:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT?*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical?*

## **IP/UDP**

### **Errors**

- IP

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:SECOnds?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:COUNt?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:RATE?*

- UDP

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:CURRent?*

## SCPI Command List - Results

### Alarms/Errors

---

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:COUNT?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:RATE?*

## ODUx

### Alarms

*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CURRent?*  
  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:CURRent?*

### Errors

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:COUNT?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:RATE?*  
  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:COUNT?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:RATE?*

### Injection

#### Mode Manual

##### - Channel

*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:ACHA*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:ACHA?*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CHAN*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CHAN?*

##### - Defect



*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE?*

**- Amount**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOUNT*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOUNT?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUNT*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUNT?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:INJECT*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:INJECT*

**Mode Rate**

**- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE?*

**- Rate**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated?*

**Mode Max Rate**

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInu*

## SCPI Command List - Results

### Alarms/Errors

---

*ous*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous?*

#### **Mode Continuous**

##### **- Channel**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:ACHA*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:ACHA?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CHAN*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CHAN?*

##### **- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE?*

##### **- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]?*

## **ODUx-TCM**

### **Alarms**

*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent?*

### **Errors**

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:COUNT?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:RATE?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:COUNT?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:RATE?*

## **Injection**

### **Mode Manual**

#### **- Defect**

*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE?*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TCM[1..n]:TY*  
*PE*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TCM[1..n]:TY*  
*PE?*

#### **- Amount**

*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOut*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOut?*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AMOut*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AMOut?*

#### **- Inject**

*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:INJect*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:INJect*

### **Mode Rate**

#### **- Defect**

*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:TYPE*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:TYPE*  
*?*  
*:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]*

## SCPI Command List - Results

### Alarms/Errors

---

*]:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n*

*]:TYPE?*

#### **- Rate**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n*

*]:RATE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n*

*]:RATE?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:RATE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:RATE*

*?*

#### **- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AUTomated*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AUTomated?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AUTomate*

*d*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AUTomate*

*d?*

#### **Mode Max Rate**

##### **- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:CON*

*Tinuous*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:CON*

*Tinuous?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n*

*]:CONTinuous*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n*

*]:CONTinuous?*

#### **Mode Continuous**

##### **- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:TYPE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]?*

**OPUx****Alarms**

*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:CURRent?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead*

**MSIM Monitoring**

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:MSIM*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:MSIM?*

**Errors**

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:RATE?*

**Injection****Channel****- Errors**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:ACHA*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:ACHA?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CHAN*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CHAN?*

## SCPI Command List - Results

### Alarms/Errors

---

#### - Alarms

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:ACHA*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:ACHA?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CHAN*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CHAN?*

#### Mode Manual

##### - Defect

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:MANual:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:MANual:TYPE?*

##### - Amount

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOut*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOut?*

##### - Inject

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:INJect*

#### Mode Rate

##### - Defect

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE?*

##### - Rate

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE?*

##### - Inject

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated?*

#### Mode Max Rate

##### - Inject

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous?*

#### Mode Continuous

##### - Defect

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E*  
**- Inject**  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:HISTory?*

## OTL

### Alarms

- Global  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:GLOBal:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:GLOBal:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:GLOBal:SECOnds?*

- Per Lane  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:SECOnds?*

### Errors

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:GLOBal:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:RATE?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:COUNt:TOTal?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:RATE:TOTal?*

### Injection

#### Lane

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:LANE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:LANE?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALANes*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALANes?*

## SCPI Command List - Results

### Alarms/Errors

---

#### **Mode Manual**

##### **- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:MANual:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:MANual:TYPE?*

##### **- Amount**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AMount*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AMount?*

##### **- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:INJect*

#### **Mode Rate**

##### **- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:TYPE?*

##### **- Rate**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:RATE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:RATE?*

##### **- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated?*

#### **Mode Max Rate**

##### **- Defect**

##### **- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:CONTInuous*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:CONTInuous?*

#### **Mode Continuous**

##### **- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALARm:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALARm:TYPE?*

##### **LOL**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm:TYPE?*



**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALARm*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ALARm?*

**LOL**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm?*

**OTUx****Alarms**

*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:SEConds?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:SEConds?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:CURRent?*

**Errors**

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:SEConds?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:COUNT?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:RATE?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:SEConds?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:COUNT?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:RATE?*

**Injection****Mode Manual****- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE?*

## SCPI Command List - Results

### Alarms/Errors

---

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE?*

**- Amount**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOut*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOut?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOut*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOut?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:INJect*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:INJect*

**Mode Rate**

**- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE?*

**- Rate**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated?*

**Mode Max Rate**

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:CONTInuous*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:CONTInuous?*

**Mode Continuous****- Defect**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]?*

**QoS Metrics****Errors**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:CURRent?*

**Section/Line/RS/MS****Alarms****- Section/RS**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:CURRent?*

**- Line/MS**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:CURRent?*

**Errors****- Section/RS**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:SECOnds?*

## SCPI Command List - Results

### Alarms/Errors

---

*:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:COUnT?*

*:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:RATE?*

- Line/MS

*:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:SEConDs?*

*:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:COUnT?*

*:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:RATE?*

### **Injection**

#### **Mode Manual**

- **Defect**

Section/RS

*:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:MANual:TYPE*

*:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:MANual:TYPE?*

Line/MS

*:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:MANual:TYPE*

*:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:MANual:TYPE?*

- **Amount**

Section/RS

*:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:AMOUNT*

*:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:AMOUNT?*

Line/MS

*:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AMOUNT*

*:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AMOUNT?*

- **Inject**

Section/RS

*:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:INJECT*

Line/MS

*:SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:INJECT*

#### **Mode Rate**

- **Defect**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTIon:AUTomated:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTIon:AUTomated:TYPE?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE?*

**- Rate**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTIon:AUTomated:RATE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTIon:AUTomated:RATE?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:RATE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:RATE?*

**- Inject**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTIon:AUTomated*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTIon:AUTomated?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated?*

**Mode Max Rate**

**- Inject**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTIon:AUTomated:CONTInuous*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTIon:AUTomated:CONTInuous*  
*?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:CONTInuous*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:CONTInuous?*

**Mode Burst Single and Burst Repeat**

**- Mode**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTIon:BURSt:MODE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTIon:BURSt:MODE?*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTIon:BURSt:MODE*

## SCPI Command List - Results

### Alarms/Errors

---

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTion:BURSt:MODE?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:MODE*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:MODE?*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:MODE*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:MODE?*

**- Defect**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:BURSt:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:BURSt:TYPE?*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTion:BURSt:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTion:BURSt:TYPE?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:TYPE?*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:TYPE?*

**- Duration**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:BURSt:DURation*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:BURSt:DURation?*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTion:BURSt:DURation*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTion:BURSt:DURation?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:DURation*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:DURation?*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:DURation*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:DURation?*

**- Period**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:BURSt:PERiod*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:SECTion:BURSt:PERiod?*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTion:BURSt:PERiod*

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTion:BURSt:PERiod?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERROR:LINE:BURSt:PERiod*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERROR:LINE:BURSt:PERiod?*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:PERiod*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt:PERiod?*

**- Inject**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERROR:SECTIon:BURSt*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERROR:SECTIon:BURSt?*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:BURSt?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERROR:LINE:BURSt*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERROR:LINE:BURSt?*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt?*

**Mode Continuous****- Defect**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon:TYPE?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:TYPE?*

**- Inject**

Section/RS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:SECTIon?*

Line/MS

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE?*

**STS-x / AU-x****Alarms**

## SCPI Command List - Results

### Alarms/Errors

---

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:CURRent?*

#### **Errors**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:RATE?*

#### **Injection**

##### **Mode Manual**

###### **- Defect**

*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE*  
*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE?*

###### **- Amount**

*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOUNT*  
*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOUNT?*

###### **- Inject**

*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJECT*

##### **Mode Rate**

###### **- Defect**

*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE*  
*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE?*

###### **- Rate**

*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE*  
*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE?*

###### **- Inject**

*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated*  
*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated?*

##### **Mode Max Rate**

###### **- Inject**

*:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:CONTInu*



*ous*

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:CONTInu  
ous?*

### **Mode Burst Single and Burst Repeat**

#### **- Mode**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:MODE  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:MODE?  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:MODE  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:MODE?*

#### **- Defect**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE?  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE?*

#### **- Duration**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:DURation  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:DURation?  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:DURation  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:DURation?*

#### **- Period**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:PERiod  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:PERiod?  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:PERiod  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:PERiod?*

#### **- Inject**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt?  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt?*

### **Mode Continuous**

#### **- Defect**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE  
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE?*

## SCPI Command List - Results

### Alarms/Errors

---

#### - Inject

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH?*

## TCM (SONET/SDH)

### Alarms

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:CURREnt?*

### Errors

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:CURREnt?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:RATE?*

### Injection

#### Mode Manual

##### - Defect

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:TYPE?*

##### - Amount

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:AMOUNT*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:AMOUNT?*

##### - Inject

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:INJECT*

#### Mode Rate

##### - Defect

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:TYPE?*

##### - Rate

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:RATE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:RATE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERROR:HOP:TCM:AUTomated*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERROR:HOP:TCM:AUTomated?*

**Mode Max Rate****- Inject**

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERROR:HOP:TCM:AUTomated:CONTInuo*  
*us*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERROR:HOP:TCM:AUTomated:CONTInuo*  
*us?*

**Mode Continuous****- Defect**

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM:TYPE?*

**- Inject**

*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM?*

## Transcoding

**Alarms**

*:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ALARm:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ALARm:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ALARm:SECOnds?*

**Errors**

*:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ERRor:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ERRor:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ERRor:SECOnds?*  
*:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ERRor:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANScode:RX:ERRor:RATE?*

**- Per Lane**

*:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:CURRent?*

## SCPI Command List - Results

### Alarms/Errors

---

*:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:SEConds?*

*:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:COUnt?*

*:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE?*

#### **- Total**

*:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:COUnt:TOTal?*

*:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE:TOTal?*

### **Injection**

#### **Lane**

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:LANE*

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:LANE?*

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:ALANes*

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:ALANes?*

#### **Mode Manual**

##### **- Defect**

Global

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:MANual:TYPE*

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:MANual:TYPE?*

Per Lane

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:MANual:TYPE*

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:MANual:TYPE?*

##### **- Amount**

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:MANual:AMOUNT*

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:MANual:AMOUNT?*

Per Lane

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:MANual:AMOUNT*

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:MANual:AMOUNT?*

##### **- Inject**

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:MANual:INJECT*

Per Lane

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:MANual:INJECT*

#### **Mode Rate**

##### **- Defect**

*:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:AUTomated:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:AUTomated:TYPE  
?*

Per Lane

*:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:AUTomated:TYPE  
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:AUTomated:TYPE?*

- **Rate**

*:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:AUTomated:RATE  
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:AUTomated:RATE  
?*

Per Lane

*:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:AUTomated:RATE  
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:AUTomated:RATE?*

- **Inject**

*:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:AUTomated  
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:AUTomated?*

Per Lane

*:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:AUTomated  
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:AUTomated?*

**Mode Max Rate**

- **Inject**

*:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:AUTomated:CON  
Tinuus  
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:AUTomated:CON  
Tinuus?*

Per Lane

*:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:AUTomated:CONTInuous  
:SOURCE[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:AUTomated:CONTInuous?*

## FTFL/PT

### FTFL

#### **Fault Indication**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication?*

#### **Code**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE?*

#### **Operator Identifier**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENtifier?*

#### **Operator Specific**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:SPECific?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:SPECific?*

### PT

Channel selection for Multi-Channel OTN

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHANnel*

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHANnel?*

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHannel*

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHannel?*

#### **Payload Type**

*:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPE:RECeived?*

*:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPE:RECeived?*

#### **Code**

*:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE:RECeived?*

*:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE:RECeived?*

**Copy RX / Global Copy RX**

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:COPY*

*:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:COPY*

*:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE*

## GFP-F/GFP-T

### Transport Layer

#### Bandwidth Usage

*:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:BANDwith:TX?*

*:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:BANDwith:RX?*

#### Mapping Efficiency

*:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:EFFiciency:TX?*

*:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:EFFiciency:RX?*

### Frame Type

*:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:COUNt:TX?*

*:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:TX?*

*:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:COUNt:RX?*

*:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:RX?*

### RX Mismatch

*:SENSe[1..n]:DATA:TELEcom:GFP:FRAMe:MISMATCH:COUNt?*

*:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel:MISMATCH:COUNt?*

### Superblock

*:FETCh[1..n]:DATA:TELEcom:GFP:SUPERblock:COUNt:TX?*

*:FETCh[1..n]:DATA:TELEcom:GFP:SUPERblock:RATE:TX?*

*:FETCh[1..n]:DATA:TELEcom:GFP:SUPERblock:COUNt:RX?*

*:FETCh[1..n]:DATA:TELEcom:GFP:SUPERblock:RATE:RX?*



## Labels

### **STS/AU Path (C2)**

*:FETCh[1..n]:DATA:TELeom:SDHSonet:HOP:PATH:LABel?*

## Logger and Alarms/Errors Logger

*:FETCh[1..n]:DATA:TELEcom:LOGGer:EVENTs?*

*:FETCh[1..n]:DATA:TELEcom:LOGGer:LIST?*

## MPLS

### Label 1/2

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:FRAMes:TX?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:FRAMes:RX?*

### Total TX/RX MPLS

#### **Ethernet BW**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:BANDwidth?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:BANDwidth?*

#### **Frame Rate**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:RATE?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:RATE?*

#### **Line Utilization**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:UTILization?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:UTILization?*

#### **Frame Count**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:COUNT?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:COUNT?*

## OTL-SDT

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:STATistics?*

### **Defect**

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:DEFect?*

### **Longest**

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LONGest?*

### **Shortest**

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:SHORtest?*

### **Last**

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LAST?*

### **Average**

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:AVERAge?*

### **Total**

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:TOTAL?*

### **Count**

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:COUNT?*

### **Longest Disruption**

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LONGest:DISRUption:DURation?*

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LONGest:DISRUption:LANE?*

### **Lanes with Disruption**

*:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LANE:DISRUption?*

## Performance Monitoring

**BERT**

*:FETCh[1..n]:DATA:TELEcom:PATtern:PM:STATistics?*

**Section/RS**

*:FETCh[1..n]:DATA:TELEcom:SONet:SECTion:PM:STATistics?*

**Line/MS**

*:FETCh[1..n]:DATA:TELEcom:SONet:LINE:PM:STATistics?*

**STS-n/AU-n**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:PM:STATistics?*

## SDT (Multi-Channel OTN)

### **Channels with Disruptions**

*:FETCh[1..n]:DATA:TELEcom:SDT:CHDisruption?*

### **Channels Monitored**

*:FETCh[1..n]:DATA:TELEcom:SDT:CHMonitored?*

### **Channels Above Threshold**

*:FETCh[1..n]:DATA:TELEcom:SDT:CHAThreshold?*

### **Longest Disruption Duration**

*:FETCh[1..n]:DATA:TELEcom:SDT:LOTImestamp?*

### **Longest Disruption Channel**

*:FETCh[1..n]:DATA:TELEcom:SDT:LOCHannel?*

### **Last Disruption Duration**

*:FETCh[1..n]:DATA:TELEcom:SDT:LATImestamp?*

### **Last Disruption Channel**

*:FETCh[1..n]:DATA:TELEcom:SDT:LACHannel?*

### **SDT Threshold**

*:SENSe[1..n]:DATA:TELEcom:SDT:THReshold*

*:SENSe[1..n]:DATA:TELEcom:SDT:THReshold?*

### Table

#### **- Longest**

*:FETCh[1..n]:DATA:TELEcom:SDT:LONGest?*

#### **- Shortest**

*:FETCh[1..n]:DATA:TELEcom:SDT:SHORtest?*

#### **- Last**

*:FETCh[1..n]:DATA:TELEcom:SDT:LAST?*

#### **- Average**

*:FETCh[1..n]:DATA:TELEcom:SDT:AVERAge?*

**- Total**

*:FETCh[1..n]:DATA:TELEcom:SDT:TOTal?*

**- Count**

*:FETCh[1..n]:DATA:TELEcom:SDT:COUNt?*

**- PASS/FAIL**

*:FETCh[1..n]:DATA:TELEcom:SDT:VERDict?*

## Service Configuration - Burst

### Committed CBS / Excess EBS

#### Frame Loss

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOSS?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOSS:VERDict?*

#### Max Jitter

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter:VERDict?*

#### Round-Trip Latency

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency:VERDict?*

#### Average RX Rate

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate:VERDict?*



## Service Configuration - Ramp

### Committed Steps

#### **TX Rate**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:TXRate?*

#### **Frame Loss**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSS?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSS:VERDict?*

#### **Max Jitter**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter:VERDict?*

#### **Round-Trip Latency**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency:VERDict?*

#### **Average RX Rate**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate:VERDict?*

## Service Performance

### SLA Parameters

#### CIR

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate?*

#### Max Jitter, Round-trip Latency, Frame Loss Rate

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VALue?*

### Metrics

#### RX Rate

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROUGHput:CURRENT?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROUGHput:CURRENT:VERDict?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROUGHput:AVERAge?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROUGHput:AVERAge:VERDict?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROUGHput:MINimum?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROUGHput:MAXimum?*

#### Jitter

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTER:CURRENT?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTER:AVERAge?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTER:MINimum?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTER:MAXimum?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTER:ESTimate?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTER:MAXimum:VERDict?*

#### Latency

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:CURRENT?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:AVERAge?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MINimum?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MAXimum?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MAXimum:VERDict?*

## Frame Loss and Out-of-Sequence

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:HISTory?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:CURRent?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:SEConds?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:COUNt?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:RATE?*

### Verdict

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:FLOSS:VERDict?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:OOSequence:VERDict?*

## Streams - Frame Loss / Out-of-Sequence

### **Frame Loss and Out-of-Sequence**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:SEConds?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:COUNt?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:RATE?*

### **Verdict**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:FLOSS:VERDict?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:OOSequence:VERDict?*

## Streams - Jitter

### **Jitter**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:AVERage?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MINimum?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MAXimum?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:ESTimate?*

### **Verdict**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MAXimum:VERDict?*

## Streams - Latency

### **Latency**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:CURRent?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:AVErAge?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MINimum?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MAXimum?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MAXimum:VERDict?*

## Streams - Throughput

### **RX Rate**

#### **Current**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:CURRent?*

#### Verdict

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:CURRent:VERDict?*

#### **Total**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:CURRent:TOTal:RXRate?*

#### **Average**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:AVERAge?*

#### Verdict

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:AVERAge:VERDict?*

#### **Minimum**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:MINimum?*

#### **Maximum**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:MAXimum?*

## Summary

### Test Status

*:FETCh[1..n]:DATA:TELEcom:TEST:STATUs?*

### Verdict

*:FETCh[1..n]:DATA:TELEcom:TEST:STATUs:VERDICT?*

### Start Time

*:FETCh[1..n]:DATA:TELEcom:TEST:STARt:TIME?*

### Test/Power Recovery

*:FETCh[1..n]:DATA:TELEcom:TEST:POWer:RECOVery:COUNt?*

## BERT

### Inject

*:SOURce[1..n]:DATA:TELEcom:PATTern:ALARm:PATTern*

*:SOURce[1..n]:DATA:TELEcom:PATTern:ALARm:PATTern?*

### Alarms

*:FETCh[1..n]:DATA:TELEcom:PATTern:ALARm:SECOnds?*

*:FETCh[1..n]:DATA:TELEcom:PATTern:ALARm:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:PATTern:ALARm:CURREnt?*

### Errors

*:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:SECOnds?*

*:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:COUNt?*

*:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:RATE?*

*:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:HISTory?*

*:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:CURREnt?*

### Verdict

*:FETCh[1..n]:DATA:TELEcom:PATTern:ERRor:PATTern:THReshold:VERDICT?*

## BER (Unframed)

### Verdict

*:FETCh[1..n]:DATA:TELEcom:UPRBs:PATTern:THReshold:VERDICT?*



**Alarms - Defect**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE?*

**Alarms - Max Range - Inject**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm*

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm?*

**Error - Manual - Defect**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:MANual:TYPE?*

**Error - Amount**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:MANual:AMOut*

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:MANual:AMOut?*

**Error - Inject - Manual**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:MANual:INJect*

**Error - Rate - Defect**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:TYPE?*

**Error - Rate**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:RATE*

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:RATE?*

**Error - Max Rate - Inject**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:CONTInuo  
us*

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:CONTInuo  
us?*

**Error - Rate - Inject**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated*

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated?*

**Lane**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:LANE*

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:LANE?*

**All**

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALANes*

*:SOURCE[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALANes?*

## Service Disruption

### Disruption Time

*:FETCh[1..n]:DATA:TELEcom:SDT:LONGest?*

*:FETCh[1..n]:DATA:TELEcom:SDT:SHORTest?*

*:FETCh[1..n]:DATA:TELEcom:SDT:LAST?*

*:FETCh[1..n]:DATA:TELEcom:SDT:AVERAge?*

*:FETCh[1..n]:DATA:TELEcom:SDT:TOTAL?*

OTL

*:FETCh[1..n]:DATA:TELEcom:OTL:SUMMery:SDT:LONGest?*

*:FETCh[1..n]:DATA:TELEcom:OTL:SUMMery:SDT:SHORTest?*

*:FETCh[1..n]:DATA:TELEcom:OTL:SUMMery:SDT:LAST?*

*:FETCh[1..n]:DATA:TELEcom:OTL:SUMMery:SDT:AVERAge?*

*:FETCh[1..n]:DATA:TELEcom:OTL:SUMMery:SDT:TOTAL?*

### Disruption Count

*:FETCh[1..n]:DATA:TELEcom:SDT:COUNt?*

OTL

*:FETCh[1..n]:DATA:TELEcom:OTL:SUMMery:SDT:COUNt?*

### Defect

*:FETCh[1..n]:DATA:TELEcom:SDT:DEFect?*

Verdict

*:FETCh[1..n]:DATA:TELEcom:SDT:VERDict?*

## Summary (EtherSAM)

### Service Configuration Test

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:STATus?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:VERDict?*

### Service Performance Test

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:STATus?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:VERDict?*

### Start Time

*:FETCh[1..n]:DATA:TELEcom:TEST:STARt:TIME?*

## Service Configuration Test

### VLAN Preservation

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:VLAN:PREServ?*

### Frame Loss

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOsS?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOsS:VERDict?*

### Max Jitter

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter:VERDict?*

*?*

### Max Latency

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency:VERDict?*

*?*

### Max RX Rate

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXRate?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXRate:VERDict?*

*?*

## Service Performance Test

### VLAN Preservation

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SERVices:VLAN:PREServ?*

### Frame Loss

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:FLOSS?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:FLOSS:VERDict?*

### Max Jitter

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:MAXJitter?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:MAXJitter:VERDict?*

*?*

### Max Latency

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:MLATency?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:MLATency:VERDict?*

*t?*

### Average RX Rate

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:ARXRate?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:ARXRate:VERDict?*

## Summary (Multi-Channel OTN)

**Test Status**

*:FETCh[1..n]:DATA:TELEcom:TEST:STATus?*

**Start Time**

*:FETCh[1..n]:DATA:TELEcom:TEST:STARt:TIME?*

**Test/Power Recovery**

*:FETCh[1..n]:DATA:TELEcom:TEST:POWer:RECOvery:COUNt?*

*Note: See Alarms/Errors for more information.*

## Summary (RFC 2544)

### Subtest Status

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TSTate?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TSTate?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSSs:TSTate?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TSTate?*

### Elapsed Time

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TETime?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TETime?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TETime?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSSs:TETime?*

### TX Frames

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:FCOunt:TX?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:FCOunt:TX?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSSs:FCOunt:TX?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:FCOunt:TX?*

### RX Frames

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:FCOunt:RX?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:FCOunt:RX?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSSs:FCOunt:RX?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:FCOunt:RX?*

### Trial #

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:CTRial?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:CTRial?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSSs:CTRial?*  
*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:CTRial?*

### Val. #

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:VALidation?*

### Step

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSSs:CSTep?*

**Start Time**

*:FETCh[1..n]:DATA:TELEcom:TEST:STARt:TIME?*

**Test/Power Recovery**

*:FETCh[1..n]:DATA:TELEcom:TEST:POWer:RECOvery:COUnT?*

**Results per frame**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THROughput:SMESsage?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:SMESsage?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:SMESsage?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:SMESsage?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THROughput:TRESults[1..n]?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:BBResults[1..n]?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:FRESult[1..n]?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:LRESults[1..n]?*

**Verdict**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:SUMMary:THREshold:VERDICT?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:THREshold:VERDICT?*

## Summary (RFC 6349)

### Test Status

*:FETCh[1..n]:DATA:TELEcom:TEST:STATUs?*

### Verdict

*:FETCh[1..n]:DATA:TELEcom:TEST:STATUs:VERDICT?*

### Start Time

*:FETCh[1..n]:DATA:TELEcom:TEST:START:TIME?*

### Test/Power Recovery

*:FETCh[1..n]:DATA:TELEcom:TEST:POWer:RECOvery:COUNt?*

### MTU

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:MTU?*

### Minimum RTT

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:MINimum:RTT?*

## Window Sweep

### Actual L4

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:L:ACTUal?*

### Verdict

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THROUGHput:ACTUal:L:VERDICT?*

## TCP Throughput

### Window

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THROUGHput:WINDow?*

### Actual L4

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THROUGHput:ACTUal:L?*

### Ideal L4

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THROUGHput:IDEal:L?*

### TCP Efficiency

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THROUGHput:TCP:EFFiciency?*



**Buffer Delay**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:BUFFer:DELay?*

**Threshold**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold?*

## Summary (Traffic Gen & Mon)

### **Test Status**

*:FETCh[1..n]:DATA:TELEcom:TEST:STATUs?*

### **Verdict**

*:FETCh[1..n]:DATA:TELEcom:TEST:STATUs:VERDICT?*

### **Start Time**

*:FETCh[1..n]:DATA:TELEcom:TEST:START:TIME?*

### **Test/Power Recovery**

*:FETCh[1..n]:DATA:TELEcom:TEST:POWER:RECOVERY:COUNt?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:STATUs:TIME?*

## Traces - OTN

### SM TTI Traces

#### SAPI

##### - Received Message

*:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:B?*

*:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:B?*

##### - Expected Message

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:EXPEcted?*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPEcted?*

##### - Copy RX

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:COPI*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:COPI*

#### DAPI

##### - Received Message

*:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:B?*

*:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:B?*

##### - Expected Message

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:EXPEcted?*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPEcted?*

##### - Copy RX

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:COPI*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:COPI*

## SCPI Command List - Results

Traces - OTN

---

### **Operator Specific**

#### **- Received Message**

*:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:OPSPec:B?*

*:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:OPSPec:B?*

#### **SAPI/DAPI OTU-TIM**

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM?*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM*

*:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM?*

### **PM TTI Traces**

Channel selection for Multi-Channel OTN

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHannel*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHannel?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHANnel*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHANnel?*

### **SAPI**

#### **- Received Message**

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:B?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:B?*

#### **- Expected Message**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPEcted?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:EXPEcted?*

### **Copy RX**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:COPIY*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:COPIY*

**DAPI****- Received Message**

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:B?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:B?*

**- Expected Message**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXPEcted?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:EXPEcted?*

**- Copy RX**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:COPIY*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:COPIY*

**Operator Specific****- Received Message**

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:OPSPec:B?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:OPSPec:B?*

**SAPI/DAPI ODU-TIM**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM?*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:TIM*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:TIM?*

**ODU - ODU TCM TTI Traces****SAPI****- Received Message**

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?*

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:B?*

**- Expected Message**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted*

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted?*

## SCPI Command List - Results

Traces - OTN

---

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:EXpected*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:EXpected?*

### **- Copy RX**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:COPI*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:COPI*

### **DAPI**

#### **- Received Message**

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:B?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:B?*

#### **- Expected Message**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXpected*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXpected?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:EXpected*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:EXpected?*

### **- Copy RX**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:COPI*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:COPI*

### **Operator Specific**

#### **- Received Message**

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:OPSPec:B?*  
*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:OPSPec:B?*

### **SAPI/DAPI TCM-TIM**

*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:TIM*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:TIM?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:TIM*  
*:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:TIM?*

## Traces - SONET/SDH

### Traces

#### Section/RS (J0)

##### - Received

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:SOVerhead:J[1..n]:TIM:PATtern:RECEived  
?*

#### STS/AU Path (J1)

##### - Received

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:POVerhead:J[1..n]:TIM:PATtern:RECEived  
?*

#### TIM-S/RS-TIM

##### - Enable

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM  
:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM?*

##### - Format

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern  
:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern?*

##### - Expected

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern:B  
:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern:B?*

##### - Copy RX

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:COPI*

#### TIM-P/HP-TIM

##### - Enable

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM  
:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM?*

##### - Format

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern  
:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern?*

## SCPI Command List - Results

Traces - SONET/SDH

---

### - **Expected**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern:B  
:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern:B?*

### - **Copy RX**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:COPI*

## TCM Access Point Identifier

### STS/AU Path (N1)

#### - **Received**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:RECeived?*

#### - **Expected**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:EXpected  
:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:EXpected?*

#### - **Enable**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:TCTim  
:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:TCTim?*

#### - **Copy RX**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:COPI*



## Traffic - Ethernet

### Total TX/RX

#### Line Utilization

*:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:LINE:UTILization?*

#### Ethernet BW

*:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:BANDwidth?*

#### Frame Rate

*:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:FRAME:RATE?*

#### Frame Count

*:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:FRAME:COUNT?*

## Frame Type

#### RX Count

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAME:COUNT:RX?*

#### TX Count

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:FRAME:COUNT:TX?*

## Frame Size

#### RX Count

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZE:COUNT?*

Total

%

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZE:PERcentage?*

## Traffic - Flow Control

### Frame Count

#### Pause Frames

*:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES?*

#### Abort Frames

*:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:ABORT?*

#### Total Frames

*:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:TX?*

*:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:RX?*

### Pause Time

*:FETCh[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSE:TIME:RX?*

### Pause Injection

#### Packet Pause Time

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSE:TIME*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSE:TIME?*

#### Inject

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSE:INJECT*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSE:INJECT*

#### Destination MAC Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRESS:ENABLE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRESS:ENABLE?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRESS*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRESS?*

# 7 *SCPI Command List - Functions*

## List of Pages

*40/100G Advanced - CFP/CFP2/CFP4/QSFP Control* on page 172

*40/100G Advanced - Lanes Mapping & Skew* on page 174

*40/100G Advanced - Pre-Emphasis* on page 175

*APS* on page 177

*Client Offset* on page 179

*Filters* on page 180

*GMP* on page 185

*OH - GFP-F/GFP-T* on page 186

*OH - OTN* on page 188

*OH - SONET/SDH* on page 189

*Packet Capture* on page 191

*Ping & Trace Route* on page 193

*Pointer Adjustment* on page 196

*RTD* on page 199

## 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

### Control Pins

Check boxes

:SOURCE[1..n]:DATA:TELEcom:CFP:CSEtting

:SOURCE[1..n]:DATA:TELEcom:CFP:CSEtting?

### Connector Power Rating

:SOURCE[1..n]:DATA:TELEcom:CFP:CPRating

:SOURCE[1..n]:DATA:TELEcom:CFP:CPRating?

### Status Pins

:SOURCE[1..n]:DATA:TELEcom:CFP:STATus?

:SENSe[1..n]:DATA:TELEcom:CFP:TX:STATus?

### MDIO Access Interface

#### MDIO Start Address

:SOURCE[1..n]:DATA:TELEcom:MDIO:START:ADDRESS

:SOURCE[1..n]:DATA:TELEcom:MDIO:START:ADDRESS?

#### MDIO End Address

:SOURCE[1..n]:DATA:TELEcom:MDIO:END:ADDRESS

:SOURCE[1..n]:DATA:TELEcom:MDIO:END:ADDRESS?

#### Bulk Read

Refer to *Bulk Read* on page 202.

#### MDIO Address

:SOURCE[1..n]:DATA:TELEcom:MDIO:ADDRESS

:SOURCE[1..n]:DATA:TELEcom:MDIO:ADDRESS?

#### MDIO Data

:SOURCE[1..n]:DATA:TELEcom:MDIO:DATA

:SOURCE[1..n]:DATA:TELEcom:MDIO:DATA?

### **Read Button**

*:SOURCE[1..n]:DATA:TELEcom:MDIO:READ*

### **Write Button**

*:SOURCE[1..n]:DATA:TELEcom:MDIO:WRITE*

## 40/100G Advanced - Lanes Mapping & Skew

### **RX Skew**

*:SENSe[1..n]:DATA:TELEcom:OTN:OTL:RX?*

### **Default/Random/Manual Mapping** button

*:SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:DEFault*

*:SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:RANDom*

*:SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:MANual*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:DEFault*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:RANDom*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:MANual*

**Note:** See Manual Mapping.

### **Reset Skew** button

*:SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:RESet*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:RESet*

**Note:** See Manual Skew.

### **Skew Alarm Threshold**

*:SOURce[1..n]:DATA:TELEcom:OTN:OTL:THReshold*

*:SOURce[1..n]:DATA:TELEcom:OTN:OTL:THReshold?*

### **Default** button

*:SOURce[1..n]:DATA:TELEcom:OTN:OTL:THReshold:DEFault*

## 40/100G Advanced - Pre-Emphasis

### All Lanes

*:SOURCE[1..n]:DATA:TELEcom:CONFig*

*:SOURCE[1..n]:DATA:TELEcom:CONFig?*

### TX

#### VOD

Per lane

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:VOD*

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:VOD?*

All lanes

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:AVOD*

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:AVOD?*

#### Pre-Emphasis Pre-Tap 0t

Per lane

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:PRET*

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:PRET?*

All lanes

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APR*

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APR?*

#### Pre-Emphasis Post-Tap 1t/2t

Per lane

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:POST[1..n]*

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:POST[1..n]?*

All lanes

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APOS[1..n]*

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APOS[1..n]?*

### RX

#### Equalizer Control

Per lane

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:ECONtrol*

## SCPI Command List - Functions

40/100G Advanced - Pre-Emphasis

---

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:ECOnTrol?*

All lanes

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:AEControl*

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:AEControl?*

### **Equalizer Gain**

Per lane

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:EGain*

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:EGain?*

All lanes

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:AEGain*

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RX:AEGain?*

### **Restore Default**

*:SOURCE[1..n]:DATA:TELEcom:TRANsceiver:RESet*



## APS

### TX

#### Switching Mode

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE?*

#### K1 Request

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINEar:REQuest*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINEar:REQuest?*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest?*

#### K1 Channel

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?*

#### K1 Destination Node Id

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:DNODE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:DNODE?*

#### K2 Protected Channel

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel?*

#### K2 Architecture

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture?*

#### K2 Operation Mode

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINEar:OMODE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINEar:OMODE?*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE?*

#### K2 Source Node Id

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE?*

**K2 Bridge Request**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest?*

**RX**

**Switching Mode**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE*  
*:SENSe[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE?*

**K1 Request**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest?*

**K1 Channel**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?*

**K1 Destination Node Id**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:DNODE?*

**K2 Protected Channel**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel?*

**K2 Architecture**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture?*

**K2 Operation Mode**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:OMODE?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE?*

**K2 Source Node Id**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE?*

**K2 Bridge Request**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest?*

## Client Offset

### TX Frequency

#### Frequency

*:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQUency?*

#### Offset

*:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQUency:ENABle*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQUency:ENABle?*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQUency:OFFSet*

*:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQUency:OFFSet?*

### RX Frequency

#### Frequency

*:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:EFREQUency?*

#### Offset, Max Negative Offset, Max Positive Offset

*:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQUency?*

#### Frequency Offset Analysis

*:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:FOANalysis:ENABle*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:FOANalysis:ENABle?*

## Filters

### Enable

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR?*

### Enabled Time

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:ENABled:TIME?*

## Filter Configuration

(  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:BRACket:OPEN*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:BRACket:OPEN?*

### Not

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:OPERator:NOT*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:OPERator:NOT?*

)  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:BRACket:CLOSe*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:BRACket:CLOSe?*

### Oper.

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:OPERator*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:OPERator?*

### Filter

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:TYPE*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:TYPE?*

### Value and Mask

#### IPv4 Destination Address

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:DESTination:IP*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:DESTination:IP?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:DESTination:IP*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:DESTination:IP?*

#### Frame Format

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:FRAMe:FORMat*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:FRAMe:FORMat?*

#### **MAC Destination Address**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:DESTination:MAC*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:DESTination:MAC?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:DESTination:MAC*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:DESTination:MAC?*

#### **UDP Destination Port**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:DESTination:UDP*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:DESTination:UDP?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:DESTination:UDP*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:DESTination:UDP?*

#### **IPv4 DiffServ**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:DSErVices*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:DSErVices?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:DSErVices*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:DSErVices?*

#### **IPv4 Precedence**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:PRECEdence*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:PRECEdence?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:PRECEdence*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:PRECEdence?*

#### **IPv4 Source Address**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:SOURce:IP*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:SOURce:IP?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:SOURce:IP*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:SOURce:IP?*

#### **MAC Source Address**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:SOURce:MAC*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:SOURce:MAC?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:SOURce:MAC*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:SOURce:MAC?*

#### **UDP Source Port**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:SOURce:UDP*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:SOURce:UDP?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FiLTeR:MASK:SOURce:UDP*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:UDP?*

#### **IPv4 TOS**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS?*

#### **S-VLAN/E-VLAN/C-VLAN ID**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:ID*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:ID?*

#### **S-VLAN/E-VLAN/C-VLAN Priority**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:PRiority*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:PRiority?*

#### **IPv4 Protocol**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:IPPRotocol*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:IPPRotocol?*

#### **EtherType**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:ETHertype*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:ETHertype?*

#### **IPv6 Destination Address**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IPVersion*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IPVersion?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IPVersion*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IPVersion?*

#### **IPv6 Source Address**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPVersion*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:SOURce:IPVersion?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:SOURce:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:SOURce:IPVersion?*

**IPv6 Flow Label**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:FLABel:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:FLABel:IPVersion?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:FLABel:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:FLABel:IPVersion?*

**IPv6 Next Header**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:NHEader:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:NHEader:IPVersion?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:NHEader:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:NHEader:IPVersion?*

**IPv6 DiffServ**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:DSERvices:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:DSERvices:IPVersion?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:DSERvices:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:DSERvices:IPVersion?*

**IPv6 Precedence**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:PRECEdence:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:PRECEdence:IPVersion?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:PRECEdence:IPVersion*  
*on*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:PRECEdence:IPVersion?*

**IPv6 TOS**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:TOS:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:TOS:IPVersion?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:TOS:IPVersion*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:TOS:IPVersion?*

**MPLS Label x**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MLABel[1..n]*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MLABel[1..n]?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:MLABel[1..n]*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel[1..n]?*

#### **MPLS COS x**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MCOS[1..n]*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MCOS[1..n]?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS[1..n]*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS[1..n]?*

## **Filter Statistics**

### **Line Utilization**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:UTILization?*

### **Ethernet BW**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:BANDwidth?*

### **Frame Rate**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:RATE?*

### **Frame Count**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:COUNT?*

### **Error Count**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:STATistics?*



## GMP

**TX Cm**

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CMStatus?*

**TX CnD**

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CNDStatus?*

**RX Cm**

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:CMStatus?*

**RX CnD**

*:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:CNDStatus?*

## OH - GFP-F/GFP-T

### TX

#### PTI

*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI*  
*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI?*

#### PFI

*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI*  
*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI?*

#### EXI

*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI*  
*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI?*

#### UPI

*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI*  
*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI?*

#### CID

*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID*  
*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID?*

#### Spare

*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe*  
*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe?*

#### Default All OH

*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:REStore:DEFAult*

#### Default (per TX overhead byte)

*:SOURCE[1..n]:DATA:TELEcom:GFP:OH:DEFAult*

### RX

#### Client Data

*:FETCh[1..n]:DATA:TELEcom:GFP:OH:DFRames?*

**Client Management**

*:FETCh[1..n]:DATA:TELEcom:GFP:OH:MFRames?*

**Reserved PTI**

*:FETCh[1..n]:DATA:TELEcom:GFP:OH:RPTiframes?*

## OH - OTN

### **OTU**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead?*  
  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead?*

### **ODU**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead?*  
  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead?*

### **OPU**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI?*  
  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:DEFault*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead?*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI?*  
*:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI?*

### **Default OTN OH**

*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:REStore:DEFault*  
  
*:SOURCE[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:DEFault*

## OH - SONET/SDH

### TX

#### Section

*:SOURCE[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead*  
*:SOURCE[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead?*  
*:SOURCE[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead:DEFault*

#### Line

*:SOURCE[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead*  
*:SOURCE[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead?*  
*:SOURCE[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead:DEFault*

#### RS

*:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]*  
*:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]?*  
*:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]:DEFault*

#### MS

*:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]*  
*:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]?*  
*:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]:DEFault*

#### STS/AU

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead?*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead:DEFault*

#### Default All OH

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:REStore:DEFault*

### RX

#### Section/RS

*:SENSe[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead?*

#### Line/MS

*:SENSe[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead?*

## SCPI Command List - Functions

*OH - SONET/SDH*

---

### **RS**

*:SENSe[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]?*

### **MS**

*:SENSe[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]?*

### **STS/AU**

*:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead?*

## Packet Capture

### Capture Source

*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:FILTer:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:FILTer:TYPE?*

### Frame Length

#### Complete/Truncated

*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:FRAMe:SIZE*  
*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:FRAMe:SIZE?*

#### Bytes

*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:BYTE*  
*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:BYTE?*

### Trigger

#### Trigger Type

*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:TSource*  
*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:TSource?*

#### On Error (error selection)

*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:TSource:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:TSource:TYPE?*

#### Trigger Position

*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:TRIGger*  
*:SOURCE[1..n]:DATA:TELEcom:CAPTURE:TRIGger?*

### Status and Controls

#### Capture

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:GLOBal:CONTRol*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:GLOBal:CONTRol?*

#### Capture Status

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STATUs?*

## SCPI Command List - Functions

### *Packet Capture*

---

#### **Frame Count**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:FRAMe:COUnT?*

#### **Buffer Usage**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:BUFFer:UTILization?*

#### **Triggered Error**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:TRIGger:ERRor?*

#### **Triggered Frame**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:CFG:STATus?*



## Ping & Trace Route

### Source IP Address

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:SOURce:IP?*

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:SOURce:IP?*

### Destination IP Address

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP*

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP?*

### Use Stream

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP:UStream*

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP:UStream?*

## Ping

### Ping button

*:SOURCE[1..n]:DATA:TELEcom:PING:SETup:RUN*

*:SOURCE[1..n]:DATA:TELEcom:PING:SETup:RUN?*

### Timeout

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TOUT*

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TOUT?*

### Delay

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:DELay*

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:DELay?*

### Data Size

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:DSIZE*

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:DSIZE?*

### TTL

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TTL*

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TTL?*

### IP TOS/DS

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TOS*

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:TOS?*

## SCPI Command List - Functions

### *Ping & Trace Route*

---

#### **Attempts**

##### **Continuous**

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:CONTInuous*  
*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:CONTInuous?*

##### **n-Attempt**

*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ATTempts*  
*:SOURCE[1..n]:DATA:TELEcom:PING:CONFig:ATTempts?*

## **Trace Route**

#### **Trace Route button**

*:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:RUN*  
*:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:RUN?*

#### **Timeout**

*:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:TOUT*  
*:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:TOUT?*

#### **Max Hop Count**

*:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:HCOunt*  
*:SOURCE[1..n]:DATA:TELEcom:TRACe:CONFig:HCOunt?*

## **Results**

#### **Ping**

*:FETCh[1..n]:DATA:TELEcom:PING:STATistics:RESults?*

#### **Trace Route**

*:FETCh[1..n]:DATA:TELEcom:TRACe:STATistics:RESults?*

## **Statistics**

#### **Packets Transmitted**

##### **Ping**

*:FETCh[1..n]:DATA:TELEcom:PING:STATistics:TX?*

##### **Trace Route**

*:FETCh[1..n]:DATA:TELEcom:TRACe:STATistics:TX?*

**Packets Received****Ping***:FETCh[1..n]:DATA:TELEcom:PING:STATistics:RX?***Trace Route***:FETCh[1..n]:DATA:TELEcom:TRACe:STATistics:TX?***Percentage Lost***:FETCh[1..n]:DATA:TELEcom:PING:STATistics:LOST?***Min Round Trip Time***:FETCh[1..n]:DATA:TELEcom:PING:STATistics:MINimum?***Max Round Trip Time***:FETCh[1..n]:DATA:TELEcom:PING:STATistics:MAXimum?***Avg. Round Trip Time***:FETCh[1..n]:DATA:TELEcom:PING:STATistics:AVERAge?*

## Pointer Adjustment

### TX Pointer Adjustment - Manual

#### Pointer Value

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:VALue?*

#### Step

##### Increment Value

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SIZE*

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SIZE?*

##### Increment button

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement*

##### Decrement Value

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SIZE*

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SIZE?*

##### Decrement button

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement*

#### Jump

##### New Pointer

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue*

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue?*

##### Inject button

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW*

##### New Data Flag

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:FLAG*

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:FLAG?*

### TX Pointer Adjustment - Sequence

#### Sequence

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:PATTern*

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:PATTern?*

**Increment/Decrement selection**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:TYPE?*

**Periodic**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:PERiodic:STATus*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:PERiodic:STATus?*

**Init-Cool**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:INITcool:STATus*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:INITcool:STATus?*

**Sequence value**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:TIMeline:VALue*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:TIMeline:VALue?*

**Sequence button**

*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter*  
*:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter?*

**Pointer Value**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:VALue?*

**Status**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:STATus?*

**RX Pointer Adjustment****Pointer Value**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:VALue?*

**Cumulative Offset**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:OFFSet?*

**Ptr. Incr.**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:COUNT?*  
*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SECOnds?*

## SCPI Command List - Functions

### *Pointer Adjustment*

---

#### **Ptr. Decr.**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:COUNt?*

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SECOnds?*

#### **NDF**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NDF:COUNt?*

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NDF:SECOnds?*

#### **No NDF**

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NNDF:COUNt?*

*:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NNDF:SECOnds?*

## RTD

### Mode

*:SENSe[1..n]:DATA:TELEcom:RTD:MODE*  
*:SENSe[1..n]:DATA:TELEcom:RTD:MODE?*

### Measure Delay button

*:SENSe[1..n]:DATA:TELEcom:RTD*  
*:SENSe[1..n]:DATA:TELEcom:RTD?*

### Status

*:FETCh[1..n]:DATA:TELEcom:RTD:DELAy:STATUs?*

### Reset button

*:SENSe[1..n]:DATA:TELEcom:RTD:RESEt*

### Delay

*:FETCh[1..n]:DATA:TELEcom:RTD:DELAy:LAST?*  
*:FETCh[1..n]:DATA:TELEcom:RTD:DELAy:MINimum?*  
*:FETCh[1..n]:DATA:TELEcom:RTD:DELAy:MAXimum?*  
*:FETCh[1..n]:DATA:TELEcom:RTD:DELAy:AVERAge?*

### Count

*:FETCh[1..n]:DATA:TELEcom:RTD:COUNt:SUCCEssful?*  
*:FETCh[1..n]:DATA:TELEcom:RTD:COUNt:FAILEd?*

## Traffic Scan

### Scan button

*:SOURCE[1..n]:DATA:TELEcom:TSCan:SCAN:ENABLE*  
*:SOURCE[1..n]:DATA:TELEcom:TSCan:SCAN:ENABLE?*

### Level

*:SOURCE[1..n]:DATA:TELEcom:TSCan:LEVel:TYPE*  
*:SOURCE[1..n]:DATA:TELEcom:TSCan:LEVel:TYPE?*

### Link Rate

*:FETCh[1..n]:DATA:TELEcom:TSCan:LINK:RATE?*

### Limit Reached

*:FETCh[1..n]:DATA:TELEcom:TSCan:LREached:STATus?*

### Discovered

*:FETCh[1..n]:DATA:TELEcom:TSCan:DISCovered?*

### Result table

*:FETCh[1..n]:DATA:TELEcom:TSCan:LIST?*

### Total Frame Count

*:FETCh[1..n]:DATA:TELEcom:TSCan:STATistics:FCOUNT:TOTAL?*

### Total Rate

*:FETCh[1..n]:DATA:TELEcom:TSCan:STATistics:RATE:TOTAL?*



## List of Pages

*Bulk Read* on page 202

*Config TCM* on page 203

*Configure Per Frame Size* on page 204

*Copy Service* on page 205

*Copy Stream* on page 206

*Filter Configuration* on page 207

*IPv6 Address Configuration* on page 210

*Laser ON/OFF Button* on page 212

*Manual Mapping* on page 213

*Manual Skew* on page 214

*Modify Frame Structure* on page 215

*Modify Tributary Slots/Port* on page 217

*Modify Trib Slots/Channels (Multi-Channel OTN)* on page 218

*Profile (Services)* on page 219

*Profile (Stream)* on page 220

*Shaping* on page 221

*Stream (Summary)* on page 222

*Thresholds (RFC 2544)* on page 223

*TOS/DS Configuration* on page 224

*Triggered Frame Details* on page 225

## **Bulk Read**

*:SOURce[1..n]:DATA:TELEcom:MDIO:BULK:READ*

*:FETCh[1..n]:DATA:TELEcom:MDIO:BULK:READ:INFormation?*

## Config TCM

TCM

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:CONFig:TCM[1..n]*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:CONFig:TCM[1..n]?*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:CONFig:TCM[1..n]*

*:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:CONFig:TCM[1..n]?*

## Configure Per Frame Size

### **All Frames**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ALL:FRAMe*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ALL:FRAMe?*

## Copy Service

*:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:COPYstream*

## Copy Stream

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:COPIstream*

## Filter Configuration

```
(
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:OPEN
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:OPEN?
```

### Not

```
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator:NOT
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator:NOT?
```

### Oper.

```
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator?
```

```
)
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:CLOSE
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:CLOSE?
```

### Filter

```
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TYPE
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TYPE?
```

### Value and Mask

#### IPv4 Destination Address

```
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IP
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IP?
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IP
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IP?
```

#### MAC Destination Address

```
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:MAC
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:MAC?
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:MAC
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:MAC?
```

#### UDP Destination Port

```
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:UDP
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:UDP?
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:UDP
:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:UDP?
```

#### **IPv4 DiffServ**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERVICES*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERVICES?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERVICES*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERVICES?*

#### **IPv4 Precedence**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECedence*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECedence?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PRECedence*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PRECedence?*

#### **IPv4 Source Address**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IP*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IP?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IP*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IP?*

#### **MAC Source Address**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:MAC*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:MAC?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:MAC*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:MAC?*

#### **UDP Source Port**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:UDP*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:UDP?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:UDP*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:UDP?*

#### **IPv4 TOS**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS?*

#### **S-VLAN/E-VLAN/C-VLAN ID**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:ID*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:ID?*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:ID*  
*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:ID?*



**S-VLAN/E-VLAN/C-VLAN Priority**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:PRiority*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:PRiority?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:PRiority*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:PRiority?*

**IPv4 Protocol**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:IPPRotocol*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:IPPRotocol?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:IPPRotocol*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:IPPRotocol?*

**EtherType**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:ETHertype*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:ETHertype?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:ETHertype*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:ETHertype?*

**MPLS Label x**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MLABel[1..n]*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MLABel[1..n]?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MLABel[1..n]*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MLABel[1..n]?*

**MPLS COS x**

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MCOS[1..n]*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MCOS[1..n]?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MCOS[1..n]*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MCOS[1..n]?*

## IPv6 Address Configuration

### Link-Local IPv6 Address

#### Mode

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAL:IPVersion:MODE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAL:IPVersion:MODE?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAL:IPVersion:MODE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAL:IPVersion:MODE?*

#### - Status

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LOCAL:IPVersion:ADDRESS:STATUS*  
*?*

#### Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAL:IPVersion:ADDRESS*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAL:IPVersion:ADDRESS?*

### Global IPv6 Address

#### Mode

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBAL:IPVersion:MODE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBAL:IPVersion:MODE?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAL:IPVersion:MODE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAL:IPVersion:MODE?*

#### Status

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBAL:IPVersion:ADDRESS:STATUS*  
*?*

#### Address

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBAL:IPVersion:ADDRESS*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBAL:IPVersion:ADDRESS?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAL:IPVersion:ADDRESS*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAL:IPVersion:ADDRESS?*

**Interface ID Coupled**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:IIcoupled*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:IIcoupled?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:IIcoupled*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:IIcoupled?*

**Prefix Mask**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:PMASK*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:PMASK?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:PMASK*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:PMASK?*

**Default Gateway****Mode**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:MODE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:MODE?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:MODE*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:MODE?*

**Status**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:DGATeway:IPVersion:ADDRes:STATus?*

**Address**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:ADDRESS*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:ADDRESS?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:ADDRESS*  
*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:ADDRESS?*

## Laser ON/OFF Button

### **Laser**

- Per lane ON/OFF

*:SENSe[1..n]:DATA:TELEcom:LASer*

*:SENSe[1..n]:DATA:TELEcom:LASer?*

- All lanes ON/OFF

*:OUTPut[1..n]:TELEcom:LASer*

*:OUTPut[1..n]:TELEcom:LASer?*

### **All Lanes selection**

*:SENSe[1..n]:DATA:TELEcom:ALASer*

*:SENSe[1..n]:DATA:TELEcom:ALASer?*

## Manual Mapping

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:TX*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:TX?*

*:SENSe[1..n]:DATA:TELEcom:ETHernet:LLAYer:RX?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX?*

## Manual Skew

### All Lanes

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane?*

### Skew value - All Lanes

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX?*

### Skew value - per Lane

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:TX*

*:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:TX?*

## Modify Frame Structure

### Global Option

#### IP Version

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion?*

### Framing

#### Frame Format

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DATAlink*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DATAlink?*

#### Network Layer

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork?*

#### Transport Layer

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANSport?*

### VLAN

#### VLAN

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACked*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACked?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STACked*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STACked?*

### **MPLS**

#### **MPLS Label**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS?*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:HEADers*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:HEADers?*



## Modify Tributary Slots/Port

### Fixed Structure

*:SOURCE[1..n]:DATA:TELEcom:OTN:FSTRUcture:ENABle*

*:SOURCE[1..n]:DATA:TELEcom:OTN:FSTRUcture:ENABle?*

### Tributary Port

*:SOURCE[1..n]:DATA:TELEcom:OTN:PORT*

*:SOURCE[1..n]:DATA:TELEcom:OTN:PORT?*

### Nominal Bit Rate

*:FETCh[1..n]:DATA:TELEcom:OTN:BITRate?*

### Number of Trib Slots

*:FETCh[1..n]:DATA:TELEcom:OTN:SLOTs?*

### Select/Un-select Slot

*:SOURCE[1..n]:DATA:TELEcom:OTN:POSition*

*:SOURCE[1..n]:DATA:TELEcom:OTN:POSition?*

### List all Slots

*:SOURCE[1..n]:DATA:TELEcom:OTN:POSition:RANGe?*

## Modify Trib Slots/Channels (Multi-Channel OTN)

### **TX = RX**

*:SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled?*

### **Assign**

*:SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries*  
*:SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries?*

### **Modify Tributary Slots**

#### **Default**

*:SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries:DEFAult*

#### **Copy RX MSI**

*:SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries:COPI*

## Profile (Services)

### **Voice / Video / Data**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE?*

### **Voice Codec**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VOICe*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VOICe?*

### **Number of Calls**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VOICe:CALLs*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VOICe:CALLs?*

### **Video Codec**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VIDEo*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VIDEo?*

### **Number of Channels**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VIDEo:CHANnels*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VIDEo:CHANnels?*

## Profile (Stream)

### Profile

#### Voice / Video / Data

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE?*

#### Voice Codec

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VOICe*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VOICe?*

#### Number of Calls

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VOICe:CALLs*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VOICe:CALLs?*

#### Video Codec

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VIDEo*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VIDEo?*

#### Number of Channels

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VIDEo:CHANnels*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VIDEo:CHANnels?*

## Shaping

### **Burst Duty Cycle**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:BANDwidth*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:BANDwidth?*

### **Period**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:TIME*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:TIME?*

### **Burst Count**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNt*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNt?*

### **Ramp Nb. of Steps**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:STEP*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:STEP?*

### **Step Time**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:TIME*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:TIME?*

### **Ramp Cycle Count**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:COUNt*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:COUNt?*

## Stream (Summary)

### **RX Frame Count**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:COUNT:FRAMES:RX?*

### **TX Frame Count**

*:FETCh[1..n]:DATA:TELEcom:ETHernet:COUNT:FRAMES:TX?*

## Thresholds (RFC 2544)

**Back-to-Back / Frame Loss / Latency / Round-Trip Latency / One-Way Latency  
Thresholds**

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:THReshold*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:REMote:THReshold?*

## TOS/DS Configuration

### Differentiated Services

#### DSCP Codepoints

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE?*

#### ECN

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS:ECN*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS:ECN?*

### Type Of Service

#### Precedence

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PREcedence*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PREcedence?*

#### Delay

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DElay*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DElay?*

#### Throughput

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:THROUGHput*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:THROUGHput?*

#### Reliability

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:RELIability*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:RELIability?*

#### Monetary Cost

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:COST*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:COST?*

#### Reserved Bit

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:BIT*

*:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:BIT?*



## Triggered Frame Details

**Frame Number**

*:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:FNUMber?*

**MAC Address**

*:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:MAC?*

*:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:MAC?*

**IP Address**

*:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:IP?*

*:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:IP?*

**Port**

*:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:PORT?*

*:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:PORT?*



This chapter presents detailed information of the SCPI commands and queries supported by the FTB/IQS-85100G.



## IMPORTANT

For SCPI commands containing the ODU[1..n] key, use:  
ODU100 for ODU0  
ODU101 for ODUflex.

For commands containing the OPU[1..n] key, use:  
OPU100 for OPU0  
OPU101 for OPUflex.

## LINStrument <LogicalInstrumentPos>

Since the platform can house many instruments, you must explicitly specify which instrument you want to remotely control.

You must add the following mnemonic at the beginning of any command or query that you send to an instrument (except for IEEE 488.2 and platform commands):

**LINStrument <LogicalInstrumentPos> :**

where <LogicalInstrumentPos> corresponds to the identification number of the instrument.

For FTB-500 and IQS-600, the identification number of the instrument is composed of the unit number (unit is always set to 1) followed by the slot number where the module is inserted to. For example, use LINS10 for a module located on slot 0.

For FTB-2 Pro, the identification number is user configurable, refer to *Remote Control - LINS* described in the platform user guide for more information.



## IMPORTANT

The parameter named BBER has been replaced by B1, B2, and B3 and will no more be accepted in the following commands:

Commands using B1:

```
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:HISTory?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:SEConds?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:CURREnt?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:COUNt?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:RATE?
```

Commands using B2:

```
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:HISTory?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:SEConds?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:CURREnt?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:COUNt?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:RATE?
```

Commands using B3:

```
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:HISTory?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:SEConds?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:CURREnt?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:COUNt?  
:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:RATE?
```

## Test Control

---

**:FETCh[1..n]:DATA:TELEcom:TEST:TIME?**

---

<b>Description</b>	<p>This query returns the time elapsed since the beginning of the test.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Time</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:TEST:TIME?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the time elapsed since the beginning of the test.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TEST ON</p> <p>FETC:DATA:TEL:TEST:TIME?</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:LOGGer:EVENTs?

---

## SCPI Command Reference

### *Test Control*

---

---

#### :FETCh[1..n]:DATA:TELEcom:MODUle:DETailS:CDATe?

---

<b>Description</b>	This query returns the Calibration Date. At *RST condition, this value is set to device-dependent. Navigation Path: About > Module Details > Calibration Date
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:MODUle:DETailS:CDATe?
<b>Response Syntax</b>	<DateTime>
<b>Response(s)</b>	<b>DateTime:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the Calibration Date.
<b>Example(s)</b>	FETC:DATA:TEL:MOD:DET:CDAT?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TIMer:DURation?

---

---

**:FETCh[1..n]:DATA:TELEcom:MODUle:DETail:SPVersion?**

---

<b>Description</b>	<p>This query returns the Software Product Version.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: About &gt; Module Details &gt; Software Product Version</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:MODUle:DETail:SPVersion?
<b>Response Syntax</b>	<Version>
<b>Response(s)</b>	<p><b>Version:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Software Product version.</p>
<b>Example(s)</b>	FETC:DATA:TEL:MOD:DET:SPV?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TEST:TYPE?

---

## SCPI Command Reference

### Test Control

---

**:SOURce[1..n]:DATA:TELEcom:VERDict:ENABLE**

<b>Description</b>	<p>This command enables/disables the test verdict status.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Test Configurator &gt; Global Options&gt;Pass/Fail verdict</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:VERDict:ENABLE &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable/disable the Verdict status.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:VERD:ENAB ON</p> <p>SOUR:DATA:TEL:VERD:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TEST:TYPE?</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:VERDict:ENABLE?**

---

<b>Description</b>	<p>This Query returns verdict status.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Test Configurator &gt;Global Options&gt; pass/Fail verdict</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:VERDict:ENABLE?
<b>Response Syntax</b>	<STATUS>
<b>Response(s)</b>	<p><b>STATUS:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status Of Verdict</p> <p>1,Verdict is enabled.</p> <p>0,Verdict is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:VERD:ENAB ON</p> <p>SOUR:DATA:TEL:VERD:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TEST:TYPE?

---

## SCPI Command Reference

### *Test Control*

---

---

**:SOURce[1..n]:DATA:TELEcom:RESTore:DEFault**

---

**Description** This command restores the values for test to default values.

**Syntax** :SOURce[1..n]:DATA:TELEcom:RESTore:DEFault

**Example(s)** SOUR:DATA:TEL:REST:DEF

**See Also** FETCh[1..n]:DATA:TELEcom:TEST:STATus?

---

---

**:SOURce[1..n]:DATA:TELEcom:TEST**

---

<b>Description</b>	<p>This command starts and stops the test.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Start</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TEST <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Starts the test.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TEST ON</p> <p>SOUR:DATA:TEL:TEST?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TEST?

---

## SCPI Command Reference

### Test Control

---

		:SOURce[1..n]:DATA:TELEcom:TEST?
<b>Description</b>	This query returns the status of test. At *RST condition, this value is set to OFF. Navigation Path: Test Setup > Start	
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TEST?	
<b>Response Syntax</b>	<Set>	
<b>Response(s)</b>	<b>Set:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the test. 1, test is ON. 0, test is OFF.	
<b>Example(s)</b>	SOUR:DATA:TEL:TEST ON SOUR:DATA:TEL:TEST? Returns: 1	
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TEST	

---

---

**:SOURce[1..n]:DATA:TELEcom:RESet**

---

<b>Description</b>	This command resets the test. This command is an event and is not associated with an *RST condition or a query form. Navigation Path: Test > Reset
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:RESet
<b>Example(s)</b>	SOUR:DATA:TEL:RES
<b>See Also</b>	SOUR:DATA:TEL:PATtern:VERDict:DISable

---

## SCPI Command Reference

### *Test Control*

---

---

#### :CONFig[1..n]:DATA:TELEcom:LOAD

---

<b>Description</b>	<p>This command loads the previously saved configuration setting as per the given path.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Load</p>
<b>Syntax</b>	:CONFig[1..n]:DATA:TELEcom:LOAD <wsp> <Path>
<b>Parameter(s)</b>	<p><b>Path:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the file name.</p>
<b>Example(s)</b>	CONF:DATA:TEL:LOAD "C:Configuration0.cfg"
<b>See Also</b>	CONFig[1..n]:DATA:TELEcom:SAVE

---

---

**:CONFig[1..n]:DATA:TELEcom:SAVE**

---

<b>Description</b>	<p>This command saves the current test configuration at a the given path.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Save</p>
<b>Syntax</b>	<code>:CONFig[1..n]:DATA:TELEcom:SAVE &lt;wsp&gt; &lt;Path&gt;</code>
<b>Parameter(s)</b>	<p><b>Path:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the file name.</p>
<b>Example(s)</b>	<code>CONF:DATA:TEL:SAVE "C:Configuration0.cfg"</code>
<b>See Also</b>	<code>CONFig[1..n]:DATA:TELEcom:LOAD</code>

---

## SCPI Command Reference

### Test Control

---

**:CONFig[1..n]:WAIT:TIME**

<b>Description</b>	<p>This command sets the wait time duration.</p> <p>Note: This command is used to insert a delay for other command. Execution of this command after any command will wait for the specified time, before taking next command to execute. Time is in milliseconds.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p>
<b>Syntax</b>	<code>:CONFig[1..n]:WAIT:TIME &lt;wsp&gt; &lt;Duration&gt;</code>
<b>Parameter(s)</b>	<p><b>Duration:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the wait time duration in milliseconds.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ITYP LANE4X10 CONF:WAIT:TIME 3000 SOUR:DATA:TEL:ITYP LANE4X25 CONF:WAIT:TIME 3000 SOUR:DATA:TEL:ITYP?</pre>
<b>See Also</b>	<code>CONFig[1..n]:DATA:TELEcom:SAVE</code>

---



**:FETCh[1..n]:DATA:TELEcom:OPTical:MODule:STATus?**

<b>Description</b>	<p>This query returns the status for optical module validation for serial CFP.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Setup &gt; CFP</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; CFP</p> <p>Navigation Path: Test Setup &gt; ETHERBERT &gt; Test Configurator &gt; Setup &gt; CFP</p> <p>Navigation Path: Test Setup &gt; RFC 2544 &gt; Test Configurator &gt; Setup &gt; CFP</p> <p>Navigation Path: Test Setup &gt; SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; CFP</p> <p>Navigation Path: Test Setup &gt; Smart Loopback &gt; Test Configurator &gt; Setup &gt; CFP</p> <p>Navigation Path: Test Setup &gt; Traffic Gen &amp; Mon &gt; Test Configurator &gt; Setup &gt; CFP</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OPTical:MODule:STATus?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Status of optical module validation for serial CFP</p> <p>NOPTicalModule, No optical module.</p> <p>VALidating, Validating.</p> <p>POWErexceed, Power Exceed.</p> <p>VALid, Valid.</p> <p>INValid, Invalid.</p> <p>UNABLEtovalidate, Unable to validate.</p> <p>MAX, MAX.</p> <p>NoOpticalAdaptor, No Optical Adaptor.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OPT:MOD:STAT?
<b>See Also</b>	FETCH[1..n]:DATA:TELEcom:MDIO:BULK:READ:INFormation?

## SCPI Command Reference

### *Test Control*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:STATus**

<b>Description</b>	<p>This command enables or disables the stream transmission.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Control &gt; TX Button</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:STATus
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables the stream transmission.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:STAT ON</p> <p>SOUR:DATA:TEL:ETH:STR:STAT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ENABLEd

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:STATus?**

<b>Description</b>	<p>This query returns the status of stream transmission.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Control &gt; TX Button</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:STATus?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of stream transmission.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:STAT ON</p> <p>SOUR:DATA:TEL:ETH:STR:STAT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled?

---

## SCPI Command Reference

### Test Control

---

**:SOURce[1..n]:DATA:TELEcom:TEST:TYPE**

<b>Description</b>	<p>This command selects the test type.</p> <p>At *RST condition, this value is set to EtherBERT.</p> <p>Navigation Path: Test Setup &gt; Test Applications &gt; Ethernet Bert</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TEST:TYPE &lt;wsp&gt;ISAM   OTNBERT   OTNSONETSDHBERT   MCOTN   SONETSDHBERT   ETHERSAM   RFC6349   RFC2544   EBERT   TMONGEN   SLOopback</p>
<b>Parameter(s)</b>	<p><b>Ttype:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: ISAM   OTNBERT   OTNSONETSDHBERT   MCOTN   SONETSDHBERT   ETHERSAM   RFC6349   RFC2544   EBERT   TMONGEN   SLOopback</p> <p>Select the test type.</p> <p>ISAM: iSAM</p> <p>OTNBERT: OTN BERT</p> <p>MCOTN: Multi-Channel OTN</p> <p>SONETSDHBERT: SONET/SDH BERT</p> <p>OTNSONETSDHBERT: OTN SONET/SDH BERT</p> <p>ETHERSAM: EtherSAM (Y.1564)</p> <p>RFC6349: RFC 6349</p> <p>RFC2544: RFC 2544</p> <p>EBERT: EtherBERT</p> <p>TMONGEN: Traffic Gen &amp; Mon</p> <p>SLOopback: Smart Loopback</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TEST:TYPE EBERT</p> <p>SOUR:DATA:TEL:TEST:TYPE?</p> <p>Returns: EBERT</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TEST:TYPE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:TEST:TYPE?**

<b>Description</b>	<p>The query returns the test type.</p> <p>At *RST condition, this value is set to EtherBERT.</p> <p>Navigation Path: Test Setup &gt; Test Applications &gt; Ethernet Bert</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TEST:TYPE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the test type.</p> <p>ISAM: iSAM</p> <p>OTNBERT: OTN BERT</p> <p>MCOTN: Multi-Channel OTN</p> <p>SONETSDHBERT: SONET/SDH BERT</p> <p>OTNSONETSDHBERT: OTN SONET/SDH BERT</p> <p>ETHERSAM: EtherSAM (Y.1564)</p> <p>RFC6349: RFC 6349</p> <p>RFC2544: RFC 2544</p> <p>EBERT: EtherBERT</p> <p>TMONGEN: Traffic Gen &amp; Mon</p> <p>SLOopback: Smart Loopback</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TEST:TYPE EBERT</p> <p>SOUR:DATA:TEL:TEST:TYPE?</p> <p>Returns: EBERT</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TEST:TYPE

---

## Modify Structure

---

**:SOURce[1..n]:DATA:TELeom:ITYPe**

---

<b>Description</b>	<p>This command sets the interface type.</p> <p>At *RST condition, this value is set to LANE10X10.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Modify Structure &gt; Interface</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELeom:ITYPe &lt;wsp&gt;LANE4X10   LANE4X25   LANE10X10   OTU3E1   OTU3E2   SERIALOC768   SERIALSTM256   SERIALOTU3   SERIALOTU3E1   SERIALOTU3E2</p>

---

:SOURce[1..n]:DATA:TELecom:ITYPe

<b>Parameter(s)</b>	<p><b>Interface:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LANE4X10   LANE4X25   LANE10X10   OTU3E1   OTU3E2   SERIALOC768   SERIALSTM256   SERIALOTU3   SERIALOTU3E1   SERIALOTU3E2</p> <p>Sets the interface type</p> <p>LANE4X10: OTU3 (4 Lanes) [43.018 Gbit/s] for transport applications and 40GE 4 lanes [41.25 Gbit/s] for datacom applications</p> <p>LANE4X25: OTU4 (4 Lanes) [111.81 Gbit/s] for transport applications and 100GE 4 lanes [103.125 Gbit/s] for datacom applications</p> <p>LANE10X10: OTU4 (10 Lanes) [111.81 Gbit/s] for transport applications and 100GE 10 lanes [103.125 Gbit/s] for datacom applications</p> <p>OTU3E1: OTU3e1 (4 Lanes) [44.571 Gbit/s]</p> <p>OTU3E2: OTU3e2 (4 Lanes) [44.583 Gbit/s]</p> <p>SERIALOC768: OC-768 [39.813 Gbit/s]</p> <p>SERIALSTM256: STM-256 [39.813 Gbit/s]</p> <p>SERIALOTU3: OTU3 [43.018 Gbit/s] serial</p> <p>SERIALOTU3E1: OTU3e1 [44.571 Gbit/s] serial</p> <p>SERIALOTU3E2: OTU3e2 [44.583 Gbit/s] serial</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ITYP LANE10X10</p> <p>SOUR:DATA:TEL:ITYP?</p> <p>Returns: LANE10X10</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:TEST

## SCPI Command Reference

### Modify Structure

---

:SOURce[1..n]:DATA:TELEcom:ITYPE?

<b>Description</b>	<p>This query returns the interface type.</p> <p>At *RST condition, this value is set to LANE10X10.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Modify Structure &gt; Interface</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ITYPE?
<b>Response Syntax</b>	<Interface>
<b>Response(s)</b>	<p><b>Interface:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the interface type.</p> <p>LANE4X10, OTU3 (4 Lanes) [43.018 Gbit/s] is selected for transport applications and 40GE 4 lanes [41.25 Gbit/s] is selected for datacom applications</p> <p>LANE4X25, OTU4 (4 Lanes) [111.81 Gbit/s] is selected for transport applications and 100GE 4 lanes [103.125 Gbit/s] is selected for datacom applications</p> <p>LANE10X10, OTU4 (10 Lanes) [111.81 Gbit/s] is selected for transport applications and 100GE 10 lanes [103.125 Gbit/s] is selected for datacom applications</p> <p>OTU3E1, OTU3e1 (4 Lanes) [44.571 Gbit/s] is selected</p> <p>OTU3E2, OTU3e2 (4 Lanes) [44.583 Gbit/s] is selected</p> <p>SERIALOC768, OC-768 [39.813 Gbit/s] is selected</p> <p>SERIALSTM256, STM-256 [39.813 Gbit/s] is selected</p> <p>SERIALOTU3, OTU3 [43.018 Gbit/s] serial is selected</p> <p>SERIALOTU3E1, OTU3e1 [44.571 Gbit/s] serial is selected</p> <p>SERIALOTU3E2, OTU3e2 [44.583 Gbit/s] serial is selected</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ITYP LANE10X10</p> <p>SOUR:DATA:TEL:ITYP?</p> <p>Returns: LANE10X10</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TEST?

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:BERT:FRAMing**

<b>Description</b>	<p>This command allows the selection of the framing type for the BERT application.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Modify Structure &gt; Framing</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:BERT:FRAMing <wsp>FRAMEDLAYER2   UNFRAMEDPCS   UNFRAMEDCAUI   UNFRAMEDXLAI
<b>Parameter(s)</b>	<p><b>Layer:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FRAMEDLAYER2   UNFRAMEDPCS   UNFRAMEDCAUI   UNFRAMEDXLAI</p> <p>Selects the framing type for BERT application.</p> <p>FRAMEDLAYER2: FRAMEDLAYER2</p> <p>UNFRAMEDPCS: UNFRAMEDPCS</p> <p>UNFRAMEDCAUI: UNFRAMEDCAUI</p> <p>UNFRAMEDXLAI: UNFRAMEDXLAI</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:BERT:FRAM FRAMEDLAYER2</p> <p>SOUR:DATA:TEL:ETH:BERT:FRAM?</p> <p>Returns: FRAMEDLAYER2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TEST:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:BERT:FRAMing?</p>

## SCPI Command Reference

### Modify Structure

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:BERT:FRAMing?

---

<b>Description</b>	<p>This query returns the framing type for the BERT application.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Modify Structure &gt; Framing</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:BERT:FRAMing?
<b>Response Syntax</b>	<Layer>
<b>Response(s)</b>	<p><b>Layer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the framing type for BERT application.</p> <p>FRAMEDLAYER2, FRAMEDLAYER2 is selected as the layer for the BERT application.</p> <p>UNFRAMEDPCS, UNFRAMEDPCS is selected as the layer for the BERT application.</p> <p>UNFRAMEDCAUI, UNFRAMEDCAUI is selected as the layer for the BERT application.</p> <p>UNFRAMEDXLAUI, UNFRAMEDXLAUI is selected as the layer for the BERT application.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:BERT:FRAM FRAMEDLAYER2</p> <p>SOUR:DATA:TEL:ETH:BERT:FRAM?</p> <p>Returns: FRAMEDLAYER2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TEST:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:BERT:FRAMing</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver**

<b>Description</b>	<p>This command selects the physical port.</p> <p>At *RST condition, this value is set to CFP.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Modify Structure &gt; Transceiver</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver <wsp>INTLBACK   CFP   ECXP   SCXP   CFP2
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: INTLBACK   CFP   ECXP   SCXP   CFP2</p> <p>Selects the physical port.</p> <p>INTLBACK: Internal Loopback.</p> <p>CFP: CFP.</p> <p>ECXP: EXFO CXP adaptor.</p> <p>SCXP: Standard CXP adaptor.</p> <p>CFP2: CFP2.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:TRAN CFP</p> <p>SOUR:DATA:TEL:ETH:PORT:TRAN?</p> <p>Returns: CFP</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver?

## SCPI Command Reference

### Modify Structure

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver?**

<b>Description</b>	<p>This query returns the selected physical port.</p> <p>At *RST condition, this value is set to CFP.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Modify Structure &gt; Transceiver</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the physical port.</p> <p>INTLBACK, Internal Loopback is selected.</p> <p>CFP, CFP is selected.</p> <p>ECXP, EXFO CXP adaptor is selected.</p> <p>SCXP, Standard CXP adaptor is selected.</p> <p>CFP2, CFP2 is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:TRAN CFP</p> <p>SOUR:DATA:TEL:ETH:PORT:TRAN?</p> <p>Returns: CFP</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:TRANsceiver

---

---

**:SOURce[1..n]:DATA:TELeom:ODU:TYPE**

<b>Description</b>	<p>This command selects the OTN Multiplexing for the Optical Data Unit (ODU).</p> <p>At *RST condition, this value is set to ODU4.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT Test Configurator &gt; Modify Structure &gt; OTN Multiplexing</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELeom:ODU:TYPE &lt;wsp&gt;O3   O3ODU2   O3O2ODU1   O3O2O1ODU0   O3ODU1   O3O1ODU0   O3ODU0   O4   O4ODU2   O4ODU0   ODU3E1   ODU3E2   O4ODU3   O4ODU1   O3GMPODUFLEX   O4GMPODUFLEX   O4ODU2E   O4GMPO3O2O1ODU0GMP   O4GMPO2O1ODU0GMP   O4GMPO2GMPODU0GMP   O4GMPODU0GMP   O3O2O1ODU0GMP   O3O1ODU0GMP   O3GMPODU0GMP   O4GMPO3O2ODU1   O4GMPO2O1ODU0   O4GMPO3O2O1ODU0   O4GmpO2GMPODU0   O4GMPO2ODU1   O4GMPO3ODU2   O4GMPO3GMPODU0   O4GMPO1ODU0   O4GMPO3ODU1   O4GMPO3GMPODU0GMP   O4GMPODU1E   O4GMPO2GMPODUFLEX   O4GMPO1ODU0GMP</pre>

---

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE

Parameter(s)	Odu:
	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p>
	<p>The allowed elements for this parameter are: O3   O3ODU2   O3O2ODU1   O3O2O1ODU0   O3ODU1   O3O1ODU0   O3ODU0   O4   O4ODU2   O4ODU0   ODU3E1   ODU3E2   O4ODU3   O4ODU1   O3GMPODUFLEX   O4GMPODUFLEX   O4ODU2E   O4GMPO3O2O1ODU0GMP   O4GMPO2O1ODU0GMP   O4GMPO2GMPODU0GMP   O4GMPODU0GMP   O3O2O1ODU0GMP   O3O1ODU0GMP   O3GMPODU0GMP   O4GMPO3O2ODU1   O4GMPO2O1ODU0   O4GMPO3O2O1ODU0   O4GmpO2GMPODU0   O4GMPO2ODU1   O4GMPO3ODU2   O4GMPO3GMPODU0   O4GMPO1ODU0   O4GMPO3ODU1   O4GMPO3GMPODU0GMP   O4GMPODU1E   O4GMPO2GMPODUFLEX   O4GMPO1ODU0GMP</p>
	<p>Selects the OTN Multiplexing for ODU (Optical Channel Data) Unit.</p>
	<p>O3, selects ODU3 for pattern client</p>
	<p>O3ODU2, selects ODU3/ODU2 for pattern and 10GbE client</p>
	<p>O3O2ODU1, selects ODU3/ODU2/ODU1 for pattern client</p>
	<p>O3O2O1ODU0, selects ODU3/ODU2/ODU1/ODU0 for pattern client</p>
	<p>O3ODU1, selects ODU3/ODU1 for pattern client</p>
	<p>O3O1ODU0, selects ODU3/ODU1/ODU0 for pattern client</p>
	<p>O3ODU0, selects ODU3/ODU0 for pattern client</p>
	<p>O4, selects ODU4 for pattern client</p>
	<p>O4ODU2, selects ODU4/ODU2 for pattern client</p>
	<p>O4ODU0, selects ODU4/ODU0 for pattern client</p>
	<p>ODU3E1, selects ODU3E1 for pattern client</p>
	<p>ODU3E2, selects ODU3E2 for pattern client</p>
	<p>O4ODU3, selects ODU4/ODU3 for pattern client</p>
	<p>O4ODU1, selects ODU4/ODU1 for pattern client</p>
	<p>O3GMPODUFLEX, selects ODU3/ODUFLEX for Ethernet (flex-GFP-F) and Pattern client</p>
	<p>O4GMPODUFLEX, selects ODU4/ODUFLEX for Ethernet (flex-GFP-F) and Pattern client</p>
	<p>O4ODU2E, selects ODU4/ODU2E for pattern client</p>
	<p>O4GMPO3O2O1ODU0GMP, selects ODU4/ODU3/ODU2/ODU1/ODU0 for 1GbE client</p>
	<p>O4GMPO2O1ODU0GMP, selects ODU4/ODU2/ODU1/ODU0 for 1GbE client</p>

:SOURCE[1..n]:DATA:TELECOM:ODU:TYPE

Parameter(s)	
	O4GMPO2GMPODU0GMP, selects ODU4/ODU2/ODU0 for 1GbE client
	O4GMPODU0GMP, selects ODU4/ODU0 for 1GbE client
	O3O2O1ODU0GMP, selects ODU3/ODU2/ODU1/ODU0 for 1GbE client
	O3O1ODU0GMP, selects ODU3/ODU1/ODU0 for 1GbE client
	O3GMPODU0GMP, selects ODU3/ODU0 for 1GbE client
	O4GMPO3O2ODU1, selects ODU4/ODU3/ODU2/ODU1 for 1GbE client
	O4GMPO2O1ODU0, selects ODU4/ODU2/ODU1/ODU0 for 1GbE client
	O4GMPO3O2O1ODU0, selects ODU4/ODU3/ODU2/ODU1/ODU0 for 1GbE client
	O4GmpO2GMPODU0, selects ODU4/ODU2/ODU0 for 1GbE client
	O4GMPO2ODU1, selects ODU4/ODU2/ODU1 for Pattern client.
	O4GMPO3Odu2, selects ODU4/ODU3/ODU3 for Pattern client.
	O4GMPO3GMPODU0, selects ODU4/ODU3/ODU0 for Pattern client.
	O4GMPO1ODU0, selects ODU4/ODU1/ODU0 for Pattern client.
	O4GMPO3ODU1, selects ODU4/ODU3/ODU1 for Pattern client.
	O4GMPO3GMPODU0GMP, selects ODU4/ODU3/ODU0 for 1GBE client.
	O4GMPODU1E, selects ODU4/ODU1E for Pattern client.
	O4GMPO2GMPODUFLEX, selects ODU4/ODU2/ODUFLEX for Ethernet (flex-GFP-F).
	O4GMPO1ODU0GMP, selects ODU4/ODU1/ODU0 for 1GBE client.
	O3Pt21Odu1, selects Odu3/Pt21/Odu1 for pattern client
	O3Pt21Odu2 ,selects Odu3/Pt21/Odu2 for pattern client
	O3Pt21O2Odu1, selects Odu3/Pt21/Odu2/Odu1 for pattern client
	O3Pt21O2Pt21Odu1,selects Odu3/Pt21/Odu2/Pt21/Odu1 for pattern client
	O3O2Pt21Odu1, selects Odu3/Odu2/Pt21/Odu1 for pattern client
	O3Pt21O1Odu0, selects Odu3/Pt21/Odu1/Odu0 for pattern client
	Odu3Pt21Odu2Odu1Odu0, selects Odu3/Pt21/Odu2/Odu1/Odu0 for pattern client
	O3Pt21O2Pt21O1Odu0, selects Odu3/Pt21/Odu2/Pt21/Odu1/Odu0 for pattern client
	O3O2Pt21O1Odu0,selects Odu3/Odu2/Pt21/Odu1/Odu0 for pattern client
	O4GmpO3Pt21O2O1Odu0,selects O4Gmp/Odu3/Pt21/Odu2/Odu1/Odu0 for pattern client
	O4GmpO3Pt21O2Pt21O1Odu0,selects O4Gmp/Odu3/Pt21/Odu2/Pt21/Odu1/Odu0 for pattern client

## SCPI Command Reference

### Modify Structure

---

**:SOURce[1..n]:DATA:TELEcom:ODU:TYPE**

<b>Parameter(s)</b>	<p>O4GmpO3O2Pt21O1Odu0,selects O4Gmp/Odu3/Odu2/Pt21/Odu1/Odu0 for pattern client</p> <p>O4GmpO3Pt21O1Odu0, selects O4Gmp/Odu3/Pt21/Odu1/Odu0 for pattern client</p> <p>O4GmpO3Pt21O2Odu1,selects O4Gmp/Odu3/Pt21/Odu2/Odu1 for pattern client</p> <p>O4GmpO3Pt21O2Pt21Odu1,selects O4Gmp/Odu3/Pt21/Odu2/Pt21/Odu1 for pattern client</p> <p>O4GmpO3O2Pt21Odu1,selects O4Gmp/Odu3/Odu2/Pt21/Odu1 for pattern client</p> <p>O4GmpO3Pt21Odu2, selects O4Gmp/Odu3/Pt21/Odu2 for pattern client</p> <p>O4GmpO2Pt21O1Odu0, selects O4Gmp/Odu2/Pt21/Odu1/Odu0 for pattern client</p> <p>O4GmpO3Pt21Odu1, selects O4Gmp/Odu3/Pt21/Odu1 for pattern client</p> <p>O4GmpO2Pt21Odu1,selects O4Gmp/Odu2/Pt21/Odu1 for pattern client</p> <p>O3Pt21O2O1Odu0Gmp,selects Odu3/Pt21/Odu2/Odu1/Odu0Gmp for pattern client</p> <p>O3Pt21O2Pt21Odu1Odu0Gmp,selects Odu3/Pt21/Odu2/Pt21/Odu1/Odu0Gmp for pattern client</p> <p>O3O2Pt21O1Odu0Gmp, selects Odu3/Odu2/Pt21/Odu1/Odu0Gmp for pattern client</p> <p>O3Pt21O1Odu0Gmp, selects Odu3/Pt21/Odu1/Odu0Gmp for pattern client</p> <p>O4GmpO3Pt21O2O1Odu0Gmp, selects O4Gmp/Odu3/Pt21/Odu2/Odu1/Odu0Gmp for 1GB client</p> <p>O4GmpO3Pt21O2Pt21O1Odu0Gmp,selects O4Gmp/Odu3/Pt21/Odu2/Pt21/Odu1/Odu0Gmp for 1GB client</p> <p>O4GmpO3O2Pt21O1Odu0Gmp, selects O4Gmp/Odu3/Odu2/Pt21/Odu1/Odu0Gmp for 1GB client</p> <p>O4GmpO2Pt21O1Odu0Gmp,selects O4Gmp/Odu2/Pt21/Odu1/Odu0Gmp for 1GB client</p> <p>O4GmpO3Pt21O1ODU0Gmp,selects O4Gmp/Odu3/Pt21/Odu1/Odu0Gmp for 1GB client</p> <p>O4GmpOdu3Gmp,selects O4Gmp/Odu3Gmp for 40GB client</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ODU:TYPE O3</p> <p>SOUR:DATA:TEL:ODU:TYPE?</p> <p>Returns: O3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ODU:TYPE?</p>



---

**:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?**

---

<b>Description</b>	This query returns the OTN Multiplexing for Optical Data Unit (ODU). At *RST condition, this value is set to ODU4. Navigation Path: Test > Setup > OTN BERT Test Configurator > Modify Structure > OTN Multiplexing
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?
<b>Response Syntax</b>	<ODUType>

---

## SCPI Command Reference

### Modify Structure

---

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?

#### Response(s)

#### ODUType:

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the OTN Multiplexing for Optical Channel Data Unit (ODU).

O3, returns ODU3 for pattern client

O3ODU2, returns ODU3/ODU2 for pattern and 10GbE client

O3O2ODU1, returns ODU3ODU2ODU1 for pattern client

O3O2O1ODU0, returns ODU3/ODU2/ODU1/ODU0 for pattern client

O3ODU1, returns ODU3/ODU1 for pattern client

O3O1ODU0, returns ODU3/ODU1/ODU0 for pattern client

O3ODU0, returns ODU3/ODU0 for pattern client

O4, returns ODU4 for pattern client

O4ODU2, returns ODU4/ODU2 for pattern client

O4ODU0, returns ODU4/ODU0 for pattern client

ODU3E1, returns ODU3E1 for pattern client

ODU3E2, returns ODU3E2 for pattern client

O4ODU3, returns ODU4/ODU3 for pattern client

O4ODU1, returns ODU4/ODU1 for pattern client

O3GMPODUFLEX, returns ODU3GMP/ODUFLEX for Ethernet (flex-GFP-F) and Pattern client

O4GMPODUFLEX, returns ODU4GMP/ODUFLEX for Ethernet (flex-GFP-F) and Pattern client

O4ODU2E, returns ODU4/ODU2E for pattern client

O4GMPO3O2O1ODU0GMP, returns ODU4/ODU3/ODU2/ODU1/ODU0 for 1GbE client

O4GMPO2O1ODU0GMP, returns ODU4/ODU2/ODU1/ODU0 for 1GbE client

O4GMPO2GMP0DU0GMP, returns ODU4/ODU2/ODU0 for 1GbE client

O4GMPODU0GMP,Returns ODU4/ODU0 for 1GbE client

O3Pt21Odu1, returns Odu3/Pt21/Odu1 for pattern client

O3Pt21Odu2 ,returns Odu3/Pt21/Odu2 for pattern client

O3Pt21O2Odu1, returns Odu3/Pt21/Odu2/Odu1 for pattern client

:SOURCE[1..n]:DATA:TELECOM:ODU:TYPE?

Response(s)	
	O3Pt21O2Pt21Odu1, returns Odu3/Pt21/Odu2/Pt21/Odu1 for pattern client
	O3O2Pt21Odu1, returns Odu3/Odu2/Pt21/Odu1 for pattern client
	O3Pt21O1Odu0, returns Odu3/Pt21/Odu1/Odu0 for pattern client
	O3Pt21O2O1Odu0, returns Odu3/Pt21/Odu2/Odu1Odu0 for pattern client
	O3Pt21O2Pt21O1Odu0, returns Odu3/Pt21/Odu2/Pt21/Odu1Odu0 for pattern client
	O3O2Pt21O1Odu0, returns Odu3/Odu2/Pt21/Odu1/Odu0 for pattern client
	O4GmpO3Pt21O2O1Odu0, returns O4Gmp/Odu3/Pt21/Odu2/Odu1/Odu0 for pattern client
	O4GmpO3Pt21O2Pt21O1Odu0, returns O4Gmp/Odu3/Pt21/Odu2/Pt21/Odu1/Odu0 for pattern client
	O4GmpO3O2Pt21Odu1Odu0, returns O4Gmp/Odu3/Odu2/Pt21/Odu1/Odu0 for pattern client
	O4GmpO3Pt21O1Odu0, returns O4Gmp/Odu3/Pt21/Odu1/Odu0 for pattern client
	O4GmpO3Pt21O2Odu1, returns O4Gmp/Odu3/Pt21/Odu2/Odu1 for pattern client
	O4GmpO3Pt21O2Pt21Odu1, returns O4Gmp/Odu3/Pt21/Odu2/Pt21/Odu1 for pattern client
	O4GmpO3O2Pt21Odu1, returns O4Gmp/Odu3/Odu2/Pt21/Odu1 for pattern client
	O4GmpO3Pt21Odu2, returns O4Gmp/Odu3/Pt21/Odu2 for pattern client
	O4GmpO2Pt21O1Odu0, returns O4Gmp/Odu2/Pt21/Odu1/Odu0 for pattern client
	O4GmpO3Pt21Odu1, returns O4Gmp/Odu3/Pt21/Odu1 for pattern client
	O4GmpO2Pt21Odu1, returns O4Gmp/Odu2/Pt21/Odu1 for pattern client
	O3Pt21O2O1Odu0Gmp, returns Odu3/Pt21/Odu2/Odu1/Odu0Gmp for pattern client
	O3Pt21O2Pt21O1Odu0Gmp, returns Odu3/Pt21/Odu2/Pt21/Odu1/Odu0Gmp for pattern client
	O3O2Pt21O1Odu0Gmp, returns Odu3/Odu2/Pt21/Odu1/Odu0Gmp for pattern client
	O3Pt21O1Odu0Gmp, returns Odu3/Pt21/Odu1/Odu0Gmp for pattern client
	O4GmpO3Pt21O2O1Odu0Gmp, returns O4Gmp/Odu3/Pt21/Odu2/Odu1/Odu0Gmp for 1GB client
	O4GmpO3Pt21O2Pt21O1Odu0Gmp, returns O4Gmp/Odu3/Pt21/Odu2/Pt21/Odu1/Odu0Gmp for 1GB client
	O4GmpO3O2Pt21O1Odu0Gmp, returns O4Gmp/Odu3/Odu2/Pt21/Odu1/Odu0Gmp for 1GB client
	O4GmpO2Pt21O1Odu0Gmp, returns O4Gmp/Odu2/Pt21/Odu1/Odu0Gmp for 1GB client

## SCPI Command Reference

### Modify Structure

---

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?

Response(s)	
	O4GmpO3Pt21O1Odu0Gmp, returns O4Gmp/Odu3/Pt21/Odu1/Odu0Gmp for 1GB client
	O4GmpOdu3Gmp, returns O4Gmp/Odu3Gmp for 40GB client
	O3O2O1ODU0GMP, returns ODU3/ODU2/ODU1/ODU0 for 1GbE client
	O3O1ODU0GMP, returns ODU3/ODU1/ODU0 for 1GbE client
	O3GMPODU0GMP, returns ODU3/ODU0 for 1GbE client
	O4GMPO3O2ODU1, returns ODU4/ODU3/ODU2/ODU1 for 1GbE client
	O4GMPO2O1ODU0, returns ODU4/ODU2/ODU1/ODU0 for 1GbE client
	O4GMPO3O2O1ODU0, returns ODU4/ODU3/ODU2/ODU1/ODU0 for 1GbE client
	O4GmpO2GMPODU0, returns ODU4/ODU2/ODU0 for 1GbE client
	O4GMPO2ODU1, returns ODU4/ODU2/ODU1 for Pattern client.
	O4GMPO3Odu2, returns ODU4/ODU3/ODU3 for Pattern client.
	O4GMPO3GMPODU0, returns ODU4/ODU3/ODU0 for Pattern client.
	O4GMPO1ODU0, returns ODU4/ODU1/ODU0 for Pattern client.
	O4GMPO3ODU1, returns ODU4/ODU3/ODU1 for Pattern client.
	O4GMPO3GMPODU0GMP, returns ODU4/ODU3/ODU0 for 1GBE client.
	O4GMPODU1E, returns ODU4/ODU1E for Pattern client.
	O4GMPO2GMPODUFLEX, returns ODU4/ODU2/ODUFLEX for Ethernet (flex-GFP-F).
	O4GMPO1ODU0GMP, returns ODU4/ODU1/ODU0 for 1GBE client.
	O3Pt21Odu1, returns Odu3/Pt21/Odu1 for pattern client
	O3Pt21Odu2 ,returns Odu3/Pt21/Odu2 for pattern client
	O3Pt21O2Odu1, returns Odu3/Pt21/Odu2/Odu1 for pattern client
	O3Pt21O2Pt21Odu1, returns Odu3/Pt21/Odu2/Pt21/Odu1 for pattern client
	O3O2Pt21Odu1, returns Odu3/Odu2/Pt21/Odu1 for pattern client
	O3Pt21O1Odu0, returns Odu3/Pt21/Odu1/Odu0 for pattern client
	Odu3Pt21Odu2Odu1Odu0, returns Odu3/Pt21/Odu2/Odu1/Odu0 for pattern client
	O3Pt21O2Pt21O1Odu0, returns Odu3/Pt21/Odu2/Pt21/Odu1/Odu0 for pattern client
	O3O2Pt21O1Odu0, returns Odu3/Odu2/Pt21/Odu1/Odu0 for pattern client
	O4GmpO3Pt21O2O1Odu0, returns O4Gmp/Odu3/Pt21/Odu2/Odu1/Odu0 for pattern client
	O4GmpO3Pt21O2Pt21O1Odu0, returns O4Gmp/Odu3/Pt21/Odu2/Pt21/Odu1/Odu0 for pattern client

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?

<b>Response(s)</b>	<p>O4GmpO3O2Pt21O1Odu0, returns O4Gmp/Odu3/Odu2/Pt21/Odu1/Odu0 for pattern client</p> <p>O4GmpO3Pt21O1Odu0, returns O4Gmp/Odu3/Pt21/Odu1/Odu0 for pattern client</p> <p>O4GmpO3Pt21O2Odu1, returns O4Gmp/Odu3/Pt21/Odu2/Odu1 for pattern client</p> <p>O4GmpO3Pt21O2Pt21Odu1, returns O4Gmp/Odu3/Pt21/Odu2/Pt21/Odu1 for pattern client</p> <p>O4GmpO3O2Pt21Odu1, returns O4Gmp/Odu3/Odu2/Pt21/Odu1 for pattern client</p> <p>O4GmpO3Pt21Odu2, returns O4Gmp/Odu3/Pt21/Odu2 for pattern client</p> <p>O4GmpO2Pt21O1Odu0, returns O4Gmp/Odu2/Pt21/Odu1/Odu0 for pattern client</p> <p>O4GmpO3Pt21Odu1, returns O4Gmp/Odu3/Pt21/Odu1 for pattern client</p> <p>O4GmpO2Pt21Odu1, returns O4Gmp/Odu2/Pt21/Odu1 for pattern client</p> <p>O3Pt21O2O1Odu0Gmp, returns Odu3/Pt21/Odu2/Odu1/Odu0Gmp for pattern client</p> <p>O3Pt21O2Pt21Odu1Odu0Gmp, returns Odu3/Pt21/Odu2/Pt21/Odu1/Odu0Gmp for pattern client</p> <p>O3O2Pt21O1Odu0Gmp, returns Odu3/Odu2/Pt21/Odu1/Odu0Gmp for pattern client</p> <p>O3Pt21O1Odu0Gmp, returns Odu3/Pt21/Odu1/Odu0Gmp for pattern client</p> <p>O4GmpO3Pt21O2O1Odu0Gmp, returns O4Gmp/Odu3/Pt21/Odu2/Odu1/Odu0Gmp for 1GB client</p> <p>O4GmpO3Pt21O2Pt21O1Odu0Gmp, returns O4Gmp/Odu3/Pt21/Odu2/Pt21/Odu1/Odu0Gmp for 1GB client</p> <p>O4GmpO3O2Pt21O1Odu0Gmp, returns O4Gmp/Odu3/Odu2/Pt21/Odu1/Odu0Gmp for 1GB client</p> <p>O4GmpO2Pt21O1Odu0Gmp, returns O4Gmp/Odu2/Pt21/Odu1/Odu0Gmp for 1GB client</p> <p>O4GmpO3Pt21O1ODU0Gmp, returns O4Gmp/Odu3/Pt21/Odu1/Odu0Gmp for 1GB client</p> <p>O4GmpOdu3Gmp, returns O4Gmp/Odu3Gmp for 40GB client</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ODU:TYPE O3</p> <p>SOUR:DATA:TEL:ODU:TYPE?</p> <p>Returns: O3</p>
<b>Note(s)</b>	<CHARACTER RESPONSE DATA>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ODU:TYPE

## SCPI Command Reference

### Modify Structure

---

**:SOURce[1..n]:DATA:TELEcom:OTN:FRAMing**

<b>Description</b>	<p>This command selects the frame layer type for OTN BERT application.</p> <p>At *RST condition, this value is set to Framed.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; Test Configurator &gt; Modify Structure &gt; Framing</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:FRAMing &lt;wsp&gt;FRAMed   10UPHLANE   20ULOGLANE</p>
<b>Parameter(s)</b>	<p><b>Layer:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FRAMed   10UPHLANE   20ULOGLANE</p> <p>Selects the frame layer type for OTN BERT application.</p> <p>FRAMed: FRAMed</p> <p>10UPHLANE: 10 Unframed Physical lanes</p> <p>20ULOGLANE: 20 Unframed Physical lanes</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:FRAM FRAMed</p> <p>SOUR:DATA:TEL:OTN:FRAM?</p> <p>Returns: FRAMed</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TEST:TYPE</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:FRAMing?**

<b>Description</b>	<p>This query returns the frame layer type for OTN BERT application.</p> <p>At *RST condition, this value is set to Framed.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Modify Structure &gt; Framing</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:FRAMing?
<b>Response Syntax</b>	<Layer>
<b>Response(s)</b>	<p><b>Layer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the frame layer type for OTN BERT application.</p> <p>FRAMed, returns FRAMed.</p> <p>10UPHLANE, returns 10 Unframed Physical lanes.</p> <p>20UOLOGLANE, returns 20 Unframed Physical lanes.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:FRAM FRAMed</p> <p>SOUR:DATA:TEL:OTN:FRAM?</p> <p>Returns: FRAMed</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TEST:TYPE?

---

## SCPI Command Reference

### Modify Structure

---

**:SOURce[1..n]:DATA:TELeom:OTN:CLient**

<b>Description</b>	<p>This command selects the client type for OTN BERT application.</p> <p>At *RST condition, this value is set to PATTERN.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Modify Structure &gt; Client</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELeom:OTN:CLient &lt;wsp&gt;PATTERN   1GBE   10GBE   40GBE   100GBE   ETHERNETFLEX</p>
<b>Parameter(s)</b>	<p><b>Client:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PATTERN   1GBE   10GBE   40GBE   100GBE   ETHERNETFLEX</p> <p>Selects the client type for OTN BERT application.</p> <p>PATTERN: Pattern</p> <p>1GBE: 1 GBE</p> <p>10GBE: 10 GBE</p> <p>40GBE: 40 GBE</p> <p>100GBE: 100 GBE</p> <p>ETHERNETFLEX: ETHERNETFLEX</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:CLI 100GBE</p> <p>SOUR:DATA:TEL:OTN:CLI?</p> <p>Returns: 100GBE</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELeom:OTN:FRAMing</p>



---

**:SOURce[1..n]:DATA:TELEcom:OTN:CLient?**

---

<b>Description</b>	<p>This query returns the client type for OTN BERT application.</p> <p>At *RST condition, this value is set to PATTERN.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Modify Structure &gt; Client</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:CLient?
<b>Response Syntax</b>	<Client >
<b>Response(s)</b>	<p><b>Client :</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the client type for OTN BERT application.</p> <p>PATTERN, Pattern as client type is selected.</p> <p>1GBE, 1GBE, as client type is selected.</p> <p>10GBE, 10GBE, as client type is selected.</p> <p>40GBE, 40GBE, as client type is selected.</p> <p>100GBE, 100GBE, as client type is selected.</p> <p>EHERNETFLEX, EHERNETFLEX as client type is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:CLI?</p> <p>Returns: PATTERN</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TAPPlicaion:TEST:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:FRAMing?</p>

---

## SCPI Command Reference

### Modify Structure

---

**:SOURce[1..n]:DATA:TELEcom:TOPology**

<b>Description</b>	<p>This command selects the topology type.</p> <p>At *RST condition, this value is set to COUPLED.</p> <p>Navigation Path: Test -&gt; Setup -&gt; Test Configurator -&gt; Modify structure -&gt; Topology</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TOPology &lt;wsp&gt;COUPLED</p>
<b>Parameter(s)</b>	<p><b>TopologyType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: COUPLED</p> <p>Selects the topology type.</p> <p>COUPLED: Coupled</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TOP COUPLED</p> <p>SOUR:DATA:TEL:TOP?</p> <p>Returns: COUPLED</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TAPplication:TEST:TYPE?</p>

---

---

**:SOURCE[1..n]:DATA:TELEcom:TOPology?**

---

<b>Description</b>	<p>The query returns the topology type.</p> <p>At *RST condition, this value is set to COUPLED.</p> <p>Navigation Path: Test(OTN BERT) -&gt; Setup -&gt; Test Configurator -&gt; Modify structure -&gt; Topology</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:TOPology?
<b>Response Syntax</b>	<TopologyType>
<b>Response(s)</b>	<p><b>TopologyType:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the topology type.</p> <p>COUPLED, returns Coupled as the topology type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TOP COUPLED</p> <p>SOUR:DATA:TEL:TOP?</p> <p>Returns: COUPLED</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:TAPplication:TEST:TYPE

---

## SCPI Command Reference

### Modify Structure

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:TRANsparent:MODE:ENABLE**

<b>Description</b>	<p>This command enables/disables the loopback mode type for Smart Loopback application.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Modify Structure &gt; Transparent</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:TRANsparent:MODE:ENABLE &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the loopback mode status for the smart loopback application.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:SLO:TRAN:MODE:ENAB ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:NETWork:VLAN:STATus</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:TRANsparent:MODE:ENABLE?**

<b>Description</b>	This query returns the status of the loopback mode type for Smart Loopback application. At *RST condition, this value is set to OFF. Navigation Path: Test > Setup > Test Configurator > Modify Structure > Transparent
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:TRANsparent:MODE:ENABLE?
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<b>set:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the loopback mode status. 1, transparent mode is enabled. 0, transparent mode is disabled.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:SLO:TRAN:MODE:ENAB?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:NETWork:VLAN:STATus?

---

## SCPI Command Reference

### *Modify Structure*

---

<b>:SOURce[1..n]:DATA:TELEcom:SONet:TEST:TYPE</b>	
<b>Description</b>	<p>This command selects the transport SONET/SDH application type.</p> <p>At *RST condition, this value is set to SONET/SDH Bert.</p> <p>Navigation Path: Test(SONET/SDH BERT) &gt; Setup &gt; Transport Test Applications</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:SONet:TEST:TYPE &lt;wsp&gt;SONETSDHBERT</code>
<b>Parameter(s)</b>	<p><b>Ttype:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SONETSDHBERT</p> <p>Selects the application type.</p> <p>SONETSDHBERT: SONETSDHBERT</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SON:TEST:TYPE SONETSDHBERT</p> <p>SOUR:DATA:TEL:SON:TEST:TYPE?</p> <p>Returns: SONETSDHBERT</p>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:TAPplication:TEST:TYPE?</code>

---

---

**:SOURce[1..n]:DATA:TELEcom:SONet:TEST:TYPE?**

---

<b>Description</b>	<p>The query returns the transport SONET/SDH application type in discrete form.</p> <p>At *RST condition, this value is set to SONET/SDH Bert.</p> <p>Navigation Path: Test (SONET/SDH BERT)&gt; Setup &gt; Transport Test Applications</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SONet:TEST:TYPE?
<b>Response Syntax</b>	<Ttype>
<b>Response(s)</b>	<p><b>Ttype:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the transport application type.</p> <p>SONETSDHBERT, SONETSDHBERT is selected as the transport SONET/SDH application type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SON:TEST:TYPE SONETSDHBERT</p> <p>SOUR:DATA:TEL:SON:TEST:TYPE?</p> <p>Returns: SONETSDHBERT</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TAPplication:TEST:TYPE

---

## SCPI Command Reference

### Modify Structure

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:FRAMing?

---

<b>Description</b>	<p>This query returns the frame layer type for SONET SDH BERT application.</p> <p>At *RST condition, this value is set to Framed.</p> <p>Navigation Path: Test ( SONET/SDH BERT)&gt; Test Configurator &gt; Modify Structure &gt; Framing</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:FRAMing?
<b>Response Syntax</b>	<Layer>
<b>Response(s)</b>	<p><b>Layer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the frame layer type for SONET/SDH BERT application.</p> <p>FRAMed, returns FRAMed as frame layer type.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDHS:FRAM?</p> <p>Returns: FRAMed</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHSonet:LINE:PM:STATus?

---



---

**:FETCh[1..n]:DATA:TELecom:SDHSonet:CLient?**

---

<b>Description</b>	<p>This query returns the client type for SONET SDH BERT application.</p> <p>At *RST condition, this value is set to Pattern.</p> <p>Navigation Path: Test ( SONET/SDH BERT)&gt; Test Configurator &gt; Modify Structure &gt; Client</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:SDHSonet:CLient?
<b>Response Syntax</b>	<Client >
<b>Response(s)</b>	<p><b>Client :</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the client type for SONET/SDH BERT application.</p> <p>PATTERN, Returns PATTERN as client type.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDHS:CLI?</p> <p>Returns: PATTERN</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:SDHSonet:FRAMing?

---

## SCPI Command Reference

### Modify Structure

---

**:SOURce[1..n]:DATA:TELEcom:HOP:TYPE**

<b>Description</b>	<p>This command selects the High Order Path (HOP) type.</p> <p>At *RST condition, the CONFig is set to a device-dependent value.</p> <p>Navigation Path: Test&gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt;Test Configurator &gt;Setup&gt;Modify Structure &gt; Sonet Multiplexing/SDH Multiplexing</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:HOP:TYPE &lt;wsp&gt;STS1   STS3C   STS12C   STS48C   STS192C   STS768C   AU3   AU4   AU44C   AU416C   AU464C   AU4256C</p>
<b>Parameter(s)</b>	<p><b>Hop:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: STS1   STS3C   STS12C   STS48C   STS192C   STS768C   AU3   AU4   AU44C   AU416C   AU464C   AU4256C</p> <p>Selects the HOP (High Order Path) type.</p> <p>STS1: SPE (Synchronous Payload Envelope) type as STS-1.</p> <p>STS3C: STS-3C.</p> <p>STS12C: STS-12C.</p> <p>STS48C: STS-48C.</p> <p>STS192C: STS-192C.</p> <p>STS768C: STS-768C.</p> <p>AU3: AU-3</p> <p>AU4: AU-4</p> <p>AU44C: AU-4-4C</p> <p>AU416C: AU-4-16C</p> <p>AU464C: AU-4-64C</p> <p>AU4256C: AU-4-256C</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:HOP:TYPE STS1</p> <p>SOUR:DATA:TEL:HOP:TYPE?</p> <p>Returns: STS1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:MULTiplex:ITYPE?</p>

---

**:SOURCE[1..n]:DATA:TELEcom:HOP:TYPE?**


---

<b>Description</b>	<p>This query returns the High Order Path (HOP) type.</p> <p>At *RST condition, the CONFig is set to a device-dependent value.</p> <p>Navigation Path: Test&gt; OTN SONET/SDH BERT OR SONET/SDH BERT&gt;Test Configurator &gt;Setup&gt;Modify Structure &gt; Sonet Multiplexing/SDH Multiplexing</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:HOP:TYPE?
<b>Response Syntax</b>	<Hop>
<b>Response(s)</b>	<p><b>Hop:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the High Order Path (HOP) type.</p> <p>NONE, No Synchronous Transport Signal Level is selected as Synchronous Payload Envelope (SPE) type.</p> <p>STS1, Synchronous Transport Signal-Level 1 (STS1) is selected as Synchronous Payload Envelope (SPE) type.</p> <p>STS3C, STS3C is selected as SPE type.</p> <p>STS12C, STS12C is selected as SPE type.</p> <p>STS48C, STS48C is selected as SPE type.</p> <p>STS192C, STS192C is selected as SPE type.</p> <p>STS768C, STS768C is selected as SPE type.</p> <p>AU3, AU-3 is selected as Administrative Unit (AU) type.</p> <p>AU4, AU-4 is selected as AU type.</p> <p>AU44C, AU-4-4C is selected as AU type.</p> <p>AU416C, AU-4-16C is selected as AU type.</p> <p>AU464C, AU-4-64C is selected as AU type.</p> <p>AU4256C, AU-4-256C is selected as AU type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:HOP:TYPE STS1</p> <p>SOUR:DATA:TEL:HOP:TYPE?</p> <p>Returns: STS1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:MULTiplex:ITYPE

---

## SCPI Command Reference

### Modify Structure

---

**:SOURce[1..n]:DATA:TELEcom:OTN:MULTIplex:ITYPe**

<b>Description</b>	<p>This command sets the interface type for the instrument.</p> <p>At *RST condition, this value is set to OC-768.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT &gt; Test Configurator &gt; Modify Structure &gt; Embedded SONET /SDH</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:MULTIplex:ITYPe &lt;wsp&gt;OC768   OC192   OC48   OC3   OC12   STM256   STM64   STM16   STM1   STM4</p>
<b>Parameter(s)</b>	<p><b>Interface:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OC768   OC192   OC48   OC3   OC12   STM256   STM64   STM16   STM1   STM4</p> <p>Sets the interface type for the instrument.</p> <p>OC768: OC-768</p> <p>OC192: OC-192</p> <p>OC48: OC-48</p> <p>OC3: OC-3</p> <p>OC12: OC-12</p> <p>STM256: STM-256</p> <p>STM64: STM-64</p> <p>STM16: STM-16</p> <p>STM1: STM-1</p> <p>STM4: STM-4</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:MULT:ITYP OC768</p> <p>SOUR:DATA:TEL:OTN:MULT:ITYP?</p> <p>Returns: OC768</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:HOP:TYPE</p>

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:MULTIplex:ITYPE?**

<b>Description</b>	<p>This query returns the interface type for the instrument.</p> <p>At *RST condition, this value is set to OC-768.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT &gt; Test Configurator &gt; Modify Structure &gt; Embedded SONET /SDH</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:MULTIplex:ITYPE?
<b>Response Syntax</b>	<Interface>
<b>Response(s)</b>	<p><b>Interface:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the interface type for the instrument.</p> <p>OC768, Returns OC768 as the interface type.</p> <p>OC192, Returns OC192 as the interface type.</p> <p>OC48, Returns OC48 as the interface type.</p> <p>OC3, Returns OC3 as the interface type.</p> <p>OC12, Returns OC12 as the interface type.</p> <p>STM256, Returns STM256 as the interface type.</p> <p>STM64, Returns STM64 as the interface type.</p> <p>STM16, Returns STM16 as the interface type.</p> <p>STM1, Returns STM1 as the interface type.</p> <p>STM4, Returns STM4 as the interface type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:MULT:ITYP OC768</p> <p>SOUR:DATA:TEL:OTN:MULT:ITYP?</p> <p>Returns: OC768</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:HOP:TYPE?

---

# Clock

**:INPut[1..n]:TELecom:BACKplane:CLOCK**

<b>Description</b>	<p>This command sets the clock mode for synchronization at the input port.</p> <p>At *RST condition, this value is set to INTERNAL.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; External/Internal/Recovered Clock &gt; Clock Mode</p>
<b>Syntax</b>	<p>:INPut[1..n]:TELecom:BACKplane:CLOCK &lt;wsp&gt;INTernal   EXTernal   Bplane   RECovered</p>
<b>Parameter(s)</b>	<p><b>Clock:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: INTernal   EXTernal   Bplane   RECovered</p> <p>Sets the clock mode for synchronization at the input port.</p> <p>INTernal, indicates the internal clock of the unit (STRATUM 3).</p> <p>EXTernal, indicates the clock received from the connected DS1/E1/2M external clock signal (port).</p> <p>Bplane, indicates the Backplane clock.</p> <p>RECovered, indicates the Recovered clock.</p>
<b>Example(s)</b>	<p>INP:TEL:BACK:CLOC INT</p> <p>INP:TEL:BACK:CLOC?</p> <p>Returns: INTernal</p>
<b>See Also</b>	<p>INPut[1..n]:TELecom:BACKplane:CLOCK?</p>

---

**:INPut[1..n]:TELEcom:BACKplane:CLOCK?**

---

<b>Description</b>	<p>This query returns the clock mode for synchronization at the input port.</p> <p>At *RST condition, this value is set to INTERNAL.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; External/Internal/Recovered Clock &gt; Clock Mode</p>
<b>Syntax</b>	:INPut[1..n]:TELEcom:BACKplane:CLOCK?
<b>Response Syntax</b>	<Clock>
<b>Response(s)</b>	<p><b>Clock:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the clock mode for synchronization at the input port.</p> <p>INTERNAL, indicates the internal clock of the unit (STRATUM 3).</p> <p>EXTERNAL, indicates the clock received from the connected DS1/E1/2M external clock signal (port).</p> <p>Bplane, indicates the Bplane clock.</p> <p>RECovered, indicates the Recovered clock.</p>
<b>Example(s)</b>	<p>INP:TEL:BACK:CLOC INT</p> <p>INP:TEL:BACK:CLOC?</p> <p>Returns: INTernal</p>
<b>See Also</b>	INPut[1..n]:TELEcom:BACKplane:CLOCK

---

## SCPI Command Reference

### Clock

---

:INPut[1..n]:TELeom:LBO

<b>Description</b>	<p>This command sets the value for the Line Build Out interface.</p> <p>At *RST condition, this value is set to DSX133.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Internal Clock &gt; LBO</p>
<b>Syntax</b>	:INPut[1..n]:TELeom:LBO <wsp>DSX655   DSX533   DSX399   DSX266   DSX133
<b>Parameter(s)</b>	<p><b>Lbo:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DSX655   DSX533   DSX399   DSX266   DSX133</p> <p>Sets the value for the Line Build Out interface.</p> <p>DSX655</p> <p>DSX533</p> <p>DSX399</p> <p>DSX266</p> <p>DSX133</p>
<b>Example(s)</b>	<p>INP:TEL:LBO DSX266</p> <p>INP:TEL:LBO?</p> <p>Returns: DSX266</p>
<b>See Also</b>	INPut[1..n]:TELeom:LEVel

---



---

**:INPut[1..n]:TELEcom:LBO?**

---

<b>Description</b>	<p>This query returns the value for the Line Build Out interface.</p> <p>At *RST condition, this value is set to DSX133.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Internal Clock &gt; LBO</p>
<b>Syntax</b>	:INPut[1..n]:TELEcom:LBO?
<b>Response Syntax</b>	<Clock>
<b>Response(s)</b>	<p><b>Clock:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the value for the Line Build Out interface.</p> <p>DSX655, sets the value as DSX655.</p> <p>DSX533, sets the value as DSX533.</p> <p>DSX399, sets the value as DSX399.</p> <p>DSX266, sets the value as DSX266.</p> <p>DSX133, sets the value as DSX133.</p>
<b>Example(s)</b>	<p>INP:TEL:LBO?</p> <p>Returns: DSX133</p>
<b>Note(s)</b>	<CHARACTER RESPONSE DATA>
<b>See Also</b>	INPut[1..n]:TELEcom:LEVel?

---

## SCPI Command Reference

### Clock

---

#### :OUTPut[1..n]:TELEcom:TERMination

<b>Description</b>	<p>This command sets the termination mode for the external timing.</p> <p>At *RST condition, this value is set to TERM.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; External Clock &gt; Termination</p>
<b>Syntax</b>	:OUTPut[1..n]:TELEcom:TERMination <wsp>TERM   MON   BRIDGE
<b>Parameter(s)</b>	<p><b>Termination:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TERM   MON   BRIDGE</p> <p>Sets the termination modes.</p> <p>TERM</p> <p>MON</p> <p>BRIDGE</p>
<b>Example(s)</b>	<p>OUTP:TERM?</p> <p>OUTP:TEL:TERM TERM</p> <p>OUTP:TEL:TERM?</p> <p>Returns: TERM</p>
<b>See Also</b>	OUTPut[1..n]:TELEcom:LEVel?

---

---

**:OUTPut[1..n]:TELEcom:TERMination?**

---

<b>Description</b>	<p>This query returns the value of the termination modes for the external timing.</p> <p>At *RST condition, this value is set to TERM.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; External Clock &gt; Termination</p>
<b>Syntax</b>	:OUTPut[1..n]:TELEcom:TERMination?
<b>Response Syntax</b>	<Termination>
<b>Response(s)</b>	<p><b>Termination:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the termination mode.</p> <p>TERM, selects Term as the termination mode.</p> <p>MON, selects MON as the termination mode.</p> <p>BRIDGE, selects BRIDGE as the termination mode.</p>
<b>Example(s)</b>	<p>OUTP:TEL:TERM</p> <p>OUTP:TEL:TERM TERM</p> <p>OUTP:TEL:TERM?</p> <p>Returns: TERM</p>
<b>See Also</b>	OUTPut[1..n]:TELEcom:LEVel

---

## SCPI Command Reference

### Clock

---

:OUTPut[1..n]:TELEcom:LEVel

<b>Description</b>	<p>This command sets the external timing interface output level for the output port. At *RST condition, this value is set to DS1. Navigation Path: Test Setup &gt; Test Configurator &gt; External Clock &gt; Interface</p>
<b>Syntax</b>	:OUTPut[1..n]:TELEcom:LEVel <wsp>2MHz   DS1   E1
<b>Parameter(s)</b>	<p><b>Level:</b> The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element. The allowed elements for this parameter are: 2MHz   DS1   E1 Selects the external timing interface output level. 2MHz: 2 MHz DS1: DS1 (Digital Signal-level 1) E1: E1</p>
<b>Example(s)</b>	<p>OUTP:TEL:LEV DS1 OUTP:TEL:LEV? Returns: DS1</p>
<b>See Also</b>	OUTPut[1..n]:TELEcom:LEVel?

---

---

**:OUTPut[1..n]:TELEcom:LEVel?**

<b>Description</b>	<p>This query returns the external timing interface output level for the output port.</p> <p>At *RST condition, this value is set to DS1.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; External Clock &gt; Interface</p>
<b>Syntax</b>	:OUTPut[1..n]:TELEcom:LEVel?
<b>Response Syntax</b>	<Level>
<b>Response(s)</b>	<p><b>Level:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the external timing interface output level.</p> <p>2MHz, selects 2 MHz as the external timing output level.</p> <p>DS1, selects DS1 (Digital Signal-level 1) as the external timing interface output level.</p> <p>E1, selects the E1 as the external timing interface.</p>
<b>Example(s)</b>	<p>OUTP:TEL:LEV DS1</p> <p>OUTP:TEL:LEV?</p> <p>Returns: DS1</p>
<b>See Also</b>	OUTPut[1..n]:TELEcom:LEVel

---

## SCPI Command Reference

### Clock

---

**:INPut[1..n]:TELeCom:LEVel**

<b>Description</b>	<p>This command sets the internal timing interface input level for the input port. At *RST condition, this value is set to DS1. Navigation Path: Test Setup &gt; Test Configurator &gt; Internal Clock &gt; Interface</p>
<b>Syntax</b>	<p>:INPut[1..n]:TELeCom:LEVel &lt;wsp&gt;2MHz   DS1   E1</p>
<b>Parameter(s)</b>	<p><b>Level:</b> The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element. The allowed elements for this parameter are: 2MHz   DS1   E1 Sets the internal timing interface input level. DS1: DS1 (Digital Signal-level 1) E1: E1 2MHZ: 2MHz</p>
<b>Example(s)</b>	<p>INP:TEL:LEV DS1 INP:TEL:LEV? Returns: DS1</p>
<b>See Also</b>	<p>INPut[1..n]:TELeCom:LEVel?</p>

---

---

**:INPut[1..n]:TELEcom:LEVel?**

<b>Description</b>	<p>This query returns the internal timing interface input level for the input port.</p> <p>At *RST condition, this value is set to DS1.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Internal Clock &gt; Interface</p>
<b>Syntax</b>	:INPut[1..n]:TELEcom:LEVel?
<b>Response Syntax</b>	<Level>
<b>Response(s)</b>	<p><b>Level:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the internal timing interface input level.</p> <p>DS1, DS1 (Digital Signal-level 1) as the interface type is selected.</p> <p>E1, E1 as the interface type is selected.</p> <p>2MHZ, 2MHZ as the interface type is selected.</p>
<b>Example(s)</b>	<p>INP:TEL:LEV DS1</p> <p>INP:TEL:LEV?</p> <p>Returns: DS1</p>
<b>See Also</b>	INPut[1..n]:TELEcom:LBO?

---

## SCPI Command Reference

### Clock

---

:OUTPut[1..n]:TELeom:FRAMing

<b>Description</b>	<p>This command selects the interface framing.</p> <p>At *RST condition, this value is set to ESF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; External Clock &gt; Framing</p>
<b>Syntax</b>	<p>:OUTPut[1..n]:TELeom:FRAMing &lt;wsp&gt;SF   ESF   PCM30   PCM30C4   PCM31   PCM31C4   SLC96</p>
<b>Parameter(s)</b>	<p><b>Framing:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SF   ESF   PCM30   PCM30C4   PCM31   PCM31C4   SLC96</p> <p>Sets the interface framing.</p> <p>SF: SF (Superframe)</p> <p>ESF: ESF (Extended Superframe)</p> <p>PCM30: PCM30 (Pulse Code Modulation)</p> <p>PCM30C4: PCM30 (Pulse Code Modulation) CRC (Cyclic Redundancy Check)</p> <p>PCM31: PCM31 (Pulse Code Modulation)</p> <p>PCM31C4: PCM31(Pulse Code Modulation) CRC (Cyclic Redundancy Check)</p> <p>SLC96: SLC96</p>
<b>Example(s)</b>	<p>OUTP:TEL:LEV DS1</p> <p>OUTP:TEL:FRAM ESF</p> <p>OUTP:TEL:FRAM?</p> <p>Returns: ESF</p>
<b>See Also</b>	<p>OUTPut[1..n]:TELeom:FRAMing?</p>



---

**:OUTPut[1..n]:TELEcom:FRAMing?**

<b>Description</b>	<p>This query returns the interface framing.</p> <p>At *RST condition, this value is set to ESF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; External Clock &gt; Framing</p>
<b>Syntax</b>	:OUTPut[1..n]:TELEcom:FRAMing?
<b>Response Syntax</b>	<Framing>
<b>Response(s)</b>	<p><b>Framing:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the interface framing.</p> <p>SF, selects SF (Superframe) as the interface framing.</p> <p>ESF, selects ESF (Extended Superframe) as the interface framing.</p> <p>PCM30, selects PCM30 (Pulse Code Modulation) as the interface framing.</p> <p>PCM30C4, selects PCM30 (Pulse Code Modulation) CRC (Cyclic Redundancy Check) as the interface framing.</p> <p>PCM31, selects PCM31 (Pulse Code Modulation) as the interface framing.</p> <p>PCM31C4, selects PCM31 (Pulse Code Modulation) CRC (Cyclic Redundancy Check) as the interface framing.</p> <p>SLC96, selects SLC96 as the interface framing.</p>
<b>Example(s)</b>	<p>OUTP:TEL:LEV DS1</p> <p>OUTP:TEL:FRAM ESF</p> <p>OUTP:TEL:FRAM?</p> <p>Returns: ESF</p>
<b>See Also</b>	OUTPut[1..n]:TELEcom:FRAMing

---

## SCPI Command Reference

### Clock

---

:INPut[1..n]:TELecom:FRAMing

<b>Description</b>	<p>This command selects the interface framing.</p> <p>At *RST condition, this value is set to ESF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Internal Clock &gt; Framing</p>
<b>Syntax</b>	<p>:INPut[1..n]:TELecom:FRAMing &lt;wsp&gt;SF   ESF   PCM30   PCM30C4   PCM31   PCM31C4   SLC96</p>
<b>Parameter(s)</b>	<p><b>Framing:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SF   ESF   PCM30   PCM30C4   PCM31   PCM31C4   SLC96</p> <p>Sets the interface framing.</p> <p>SF: SF (Superframe)</p> <p>ESF: ESF (Extended Superframe)</p> <p>PCM30: PCM30 (Pulse Code Modulation)</p> <p>PCM30C4: PCM30 (Pulse Code Modulation) CRC (Cyclic Redundancy Check)</p> <p>PCM31: PCM31 (Pulse Code Modulation)</p> <p>PCM31C4: PCM31(Pulse Code Modulation) CRC (Cyclic Redundancy Check)</p> <p>SLC96: SLC96</p>
<b>Example(s)</b>	<p>INP:TEL:LEV DS1</p> <p>INP:TEL:FRAM ESF</p> <p>INP:TEL:FRAM?</p> <p>Returns: ESF</p>
<b>See Also</b>	<p>INPut[1..n]:TELecom:FRAMing?</p>

---

**:INPut[1..n]:TELEcom:FRAMing?**

<b>Description</b>	<p>This query returns the interface framing.</p> <p>At *RST condition, this value is set to ESF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Internal Clock &gt; Framing</p>
<b>Syntax</b>	:INPut[1..n]:TELEcom:FRAMing?
<b>Response Syntax</b>	<Framing>
<b>Response(s)</b>	<p><b>Framing:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the interface framing.</p> <p>SF, selects SF (Superframe) as the interface framing.</p> <p>ESF, selects ESF (Extended Superframe) as the interface framing.</p> <p>PCM30, selects PCM30 (Pulse Code Modulation) as the interface framing.</p> <p>PCM30C4, selects PCM30 (Pulse Code Modulation) CRC (Cyclic Redundancy Check) as the interface framing.</p> <p>PCM31, selects PCM31 (Pulse Code Modulation) as the interface framing.</p> <p>PCM31C4, selects PCM31(Pulse Code Modulation) CRC (Cyclic Redundancy Check) as the interface framing.</p> <p>SLC96, selects SLC96 as the interface framing.</p>
<b>Example(s)</b>	<p>INP:TEL:LEV DS1</p> <p>INP:TEL:FRAM ESF</p> <p>INP:TEL:FRAM?</p> <p>Returns: ESF</p>
<b>Note(s)</b>	<CHARACTER RESPONSE DATA>
<b>See Also</b>	INPut[1..n]:TELEcom:FRAMing

---

## SCPI Command Reference

### Clock

---

:OUTPut[1..n]:TELecom:CODE

<b>Description</b>	<p>This command sets the external timing line code for the output port.</p> <p>At *RST condition, this value is set to B8ZS.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; External Clock &gt; Line Coding</p>
<b>Syntax</b>	:OUTPut[1..n]:TELecom:CODE <wsp>B8ZS   HDB3   AMI
<b>Parameter(s)</b>	<p><b>Code:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B8ZS   HDB3   AMI</p> <p>Sets the external timing line code.</p> <p>B8ZS</p> <p>HDB3</p> <p>AMI</p>
<b>Example(s)</b>	<p>OUTP:TEL:LEV E1</p> <p>OUTP:TEL:CODE HDB3</p> <p>OUTP:TEL:CODE?</p> <p>Returns: HDB3</p>
<b>See Also</b>	OUTPut[1..n]:TELecom:CODE?

---

---

**:OUTPut[1..n]:TELEcom:CODE?**

<b>Description</b>	<p>This query returns the external timing line code for the output port.</p> <p>At *RST condition, this value is set to B8ZS.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; External Clock &gt; Line Coding</p>
<b>Syntax</b>	:OUTPut[1..n]:TELEcom:CODE?
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the external timing line code.</p> <p>B8ZS, B8ZS external timing line code is selected.</p> <p>HDB3, HDB3 external timing line code is selected.</p> <p>AMI, AMI external timing line code is selected.</p>
<b>Example(s)</b>	<p>OUTP:TEL:LEV E1</p> <p>OUTP:TEL:CODE HDB3</p> <p>OUTP:TEL:CODE?</p> <p>Returns: HDB3</p>
<b>See Also</b>	OUTPut[1..n]:TELEcom:CODE

---

## SCPI Command Reference

### Clock

---

**:INPut[1..n]:TELecom:CODE**

<b>Description</b>	<p>This command sets the interface line coding for the input port.</p> <p>At *RST condition, this value is set to B8ZS.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Internal Clock &gt; Line Coding</p>
<b>Syntax</b>	<code>:INPut[1..n]:TELecom:CODE &lt;wsp&gt;B8ZS   HDB3   AMI</code>
<b>Parameter(s)</b>	<p><b>Code:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B8ZS   HDB3   AMI</p> <p>Sets the interface line coding.</p> <p>B8ZS</p> <p>HDB3</p> <p>AMI</p>
<b>Example(s)</b>	<p>INP:TEL:LEV E1</p> <p>INP:TEL:CODE HDB3</p> <p>INP:TEL:CODE?</p> <p>Returns: HDB3</p>
<b>See Also</b>	<code>INPut[1..n]:TELecom:CODE?</code>

---

---

**:INPut[1..n]:TELeom:CODE?**

<b>Description</b>	<p>This query returns the interface line coding for the input port.</p> <p>At *RST condition, this value is set to B8ZS.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Internal Clock &gt; Line Coding</p>
<b>Syntax</b>	:INPut[1..n]:TELeom:CODE?
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the interface line coding.</p> <p>B8ZS, selects the B8ZS as the interface line coding.</p> <p>HDB3, selects the HDB3 as the interface line coding.</p> <p>AMI, selects the AMI as the interface line coding.</p>
<b>Example(s)</b>	<p>INP:TEL:LEV E1</p> <p>INP:TEL:CODE HDB3</p> <p>INP:TEL:CODE?</p> <p>Returns: HDB3</p>
<b>See Also</b>	INPut[1..n]:TELeom:CODE

---

## SCPI Command Reference

### Clock

---

**:OUTPut[1..n]:TELEcom:CLOCK:FREQuency?**

<b>Description</b>	This query returns the received frequency signal rate. At *RST condition, this value is set to device-dependent. Navigation Path: Test Setup > Test Configurator > External Clock > Frequency
<b>Syntax</b>	:OUTPut[1..n]:TELEcom:CLOCK:FREQuency?
<b>Response Syntax</b>	<Frequency>
<b>Response(s)</b>	<b>Frequency:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the received signal rate.
<b>Example(s)</b>	OUTP:TEL:LEV DS1 OUTP:TEL:CLOC:FREQ?
<b>See Also</b>	OUTPut[1..n]:TELEcom:CLOCK:FREQuency:OFFSet?

---



---

**:OUTPut[1..n]:TELEcom:CLOCK:FREQuency:OFFSet?**

<b>Description</b>	<p>This query returns the positive or negative frequency offset between the standard rate specification and the rate from the received signal.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; External Clock &gt; Offset</p>
<b>Syntax</b>	:OUTPut[1..n]:TELEcom:CLOCK:FREQuency:OFFSet?
<b>Response Syntax</b>	<Offset>
<b>Response(s)</b>	<p><b>Offset:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frequency offset.</p>
<b>Example(s)</b>	<p>OUTP:TEL:LEV DS1</p> <p>OUTP:TEL:CLOC:FREQ:OFFS?</p>
<b>See Also</b>	OUTPut[1..n]:TELEcom:CLOCK:FREQuency?

---

## SCPI Command Reference

### Clock

---

---

#### :INPut[1..n]:TELEcom:COUtput:FREQuency?

---

<b>Description</b>	<p>This query returns the frequency of clock output.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; External Clock &gt; Clock Output &gt; Frequency</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Internal Clock &gt; Clock Output &gt; Frequency</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Backplane &gt; Clock Output &gt; Frequency</p>
<b>Syntax</b>	:INPut[1..n]:TELEcom:COUtput:FREQuency?
<b>Response Syntax</b>	<Frequency>
<b>Response(s)</b>	<p><b>Frequency:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the offset value of the frequency.</p>
<b>Example(s)</b>	INP:TEL:COU:FREQ?
<b>See Also</b>	INPut[1..n]:TELEcom:COUtput:STAtus?

---

---

**:INPut[1..n]:TELEcom:COUtput:STATus?**

---

<b>Description</b>	<p>This query returns the current state of clock output.</p> <p>This query is not associated with any *RST value.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; External Clock &gt; Clock Output &gt; Clock Out</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Internal Clock &gt; Clock Output &gt; Clock Out</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Backplane &gt; Clock Output &gt; Clock Out</p>
<b>Syntax</b>	:INPut[1..n]:TELEcom:COUtput:STATus?
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current state of the clock out field.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	INP:TEL:COU:STAT?
<b>See Also</b>	INPut[1..n]:TELEcom:COUtput:FREQuency?

---

## SCPI Command Reference

### Clock

---

**:INPut[1..n]:TELecom:BCLock:ENABle**

<b>Description</b>	<p>This command sets the backplane clock enabled status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; External Clock &gt; Backplane &gt; Backplane Clock</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Internal Clock &gt; Backplane &gt; Backplane Clock</p>
<b>Syntax</b>	<p>:INPut[1..n]:TELecom:BCLock:ENABle &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the backplane clock enabled status.</p>
<b>Example(s)</b>	<p>INP:TEL:BCL:ENAB ON</p> <p>INP:TEL:BCL:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>INPut[1..n]:TELecom:BACKplane:STAtus?</p>

---

**:INPut[1..n]:TELecom:BCLock:ENABLE?**

---

<b>Description</b>	<p>This query returns the backplane clock value of the backplane.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; External Clock &gt; Backplane &gt; Backplane Clock</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Internal Clock &gt; Backplane &gt; Backplane Clock</p>
<b>Syntax</b>	:INPut[1..n]:TELecom:BCLock:ENABLE?
<b>Response Syntax</b>	<Clock>
<b>Response(s)</b>	<p><b>Clock:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the clock value of the backplane.</p> <p>1, backplane clock is enabled.</p> <p>0, backplane clock is disabled.</p>
<b>Example(s)</b>	<p>INP:TEL:BCL:ENAB ON</p> <p>INP:TEL:BCL:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	INPut[1..n]:TELecom:BACKplane:STAtus?

---

## SCPI Command Reference

### Clock

---

#### :INPut[1..n]:TELEcom:BACKplane:STATus?

<b>Description</b>	<p>This query returns the current state of the backplane.</p> <p>This command is not associated with any *RST value.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; External Clock &gt; Backplane &gt; Backplane Clock</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Internal Clock &gt; Backplane &gt; Backplane Clock</p>
<b>Syntax</b>	:INPut[1..n]:TELEcom:BACKplane:STATus?
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current Backplane status.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	INP:TEL:BACK:STAT?
<b>See Also</b>	INPut[1..n]:TELEcom:BCLock:ENABle?

**:INPut[1..n]:TELEcom:CLOCK:ALARm:STATus?**

<b>Description</b>	<p>This query returns the status Clock out.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Internal Clock &gt; Ext Clock Out &gt; Clock Out</p>
<b>Syntax</b>	:INPut[1..n]:TELEcom:CLOCK:ALARm:STATus? <wsp>LOC
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOC</p> <p>Sets the status clock Out.</p> <p>LOC</p>
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the status of Clock out.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	INP:TEL:CLOC:ALAR:STAT? LOC
<b>See Also</b>	OUTPut[1..n]:TELEcom:CLOCK:ALARm:STATus?

## SCPI Command Reference

### Clock

---

**:OUTPut[1..n]:TELEcom:CLOCK:ALARm:STATus?**

<b>Description</b>	<p>This query returns the status Clock in.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; External Clock &gt; Ext Clock In &gt; Clock In</p>
<b>Syntax</b>	<p>:OUTPut[1..n]:TELEcom:CLOCK:ALARm:STATus? &lt;wsp&gt;LOS   LOF   AIS   FREQUENCY</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOS   LOF   AIS   FREQUENCY</p> <p>Sets the status clock in.</p> <p>LOS</p> <p>LOF</p> <p>AIS</p> <p>FREQUENCY</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the status of Clock out.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>OUTP:TEL:CLOC:ALAR:STAT? LOS</p>
<b>See Also</b>	<p>OUTPut[1..n]:TELEcom:BACKplane:STATus?</p>



## CFP/CFP2

---

:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:MODule:ID?

---

<b>Description</b>	<p>This query returns the module id of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:MODule:ID?
<b>Response Syntax</b>	<Module_id>
<b>Response(s)</b>	<p><b>Module_id:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the module id of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:MOD:ID?
<b>See Also</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:VENDor:NAME?

---

## SCPI Command Reference

### CFP/CFP2

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:CFP:VENDor:NAME?

---

<b>Description</b>	<p>This query returns the vendor name of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:VENDor:NAME?
<b>Response Syntax</b>	<Vendor >
<b>Response(s)</b>	<p><b>Vendor :</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the vendor name of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:VEND:NAME?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:CFP:PART:NUMBer?

---

---

**:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:PART:NUMBer?**

---

<b>Description</b>	<p>This query returns the part number of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:PART:NUMBer?
<b>Response Syntax</b>	<Part_Number>
<b>Response(s)</b>	<p><b>Part_Number:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the part number of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:PART:NUMB?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:CFP:SERial:NUMBer?

---

## SCPI Command Reference

### CFP/CFP2

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:CFP:SERial:NUMBER?

---

<b>Description</b>	<p>This query returns the serial number of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:SERial:NUMBER?
<b>Response Syntax</b>	<Serial_Number>
<b>Response(s)</b>	<p><b>Serial_Number:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the serial number of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:SER:NUMB?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:CFP:REVIion?

---

---

**:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:REVision?**

---

<b>Description</b>	<p>This query returns the revision of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:REVision?
<b>Response Syntax</b>	<Revision>
<b>Response(s)</b>	<p><b>Revision:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the revision of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:REV?
<b>See Also</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:CONNeCtor:TYPE?

---

## SCPI Command Reference

### CFP/CFP2

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:CFP:FIRMware:VERSion?

---

<b>Description</b>	<p>This query returns the firmware version of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:FIRMware:VERSion?
<b>Response Syntax</b>	<Firmware_Version>
<b>Response(s)</b>	<p><b>Firmware_Version:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the firmware version of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:FIRM:VERS?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:CFP:CONNector:TYPE?

---

**:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:CONNector:TYPE?**

<b>Description</b>	<p>This query returns the connector type of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:CONNector:TYPE?
<b>Response Syntax</b>	<Connector_Type>
<b>Response(s)</b>	<p><b>Connector_Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the connector type of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:CONN:TYPE?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:CFP:SPEed?

---

## SCPI Command Reference

### CFP/CFP2

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:CFP:SPEEd?

---

<b>Description</b>	<p>This query returns the speed of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:SPEEd?
<b>Response Syntax</b>	<Speed>
<b>Response(s)</b>	<p><b>Speed:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the speed of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:SPE?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:CFP:TYPE?

---



---

**:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:TYPE?**

---

<b>Description</b>	<p>This query returns the type of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:TYPE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the type of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:TYPE?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:CFP:POWer:CLASs?

---

## SCPI Command Reference

### CFP/CFP2

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:CFP:POWer:CLASs?

---

<b>Description</b>	<p>This query returns the power class of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:POWer:CLASs?
<b>Response Syntax</b>	<Power_Class>
<b>Response(s)</b>	<p><b>Power_Class:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the power class of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:POW:CLAS?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:CFP:LRATio:TYPE?

---

---

**:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:LRATio:TYPE?**

<b>Description</b>	<p>This query returns the lane ratio of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:LRATio:TYPE?
<b>Response Syntax</b>	<Lane_Ratio>
<b>Response(s)</b>	<p><b>Lane_Ratio:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the lane ratio of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:LRAT:TYPE?
<b>See Also</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:WDM:TYPE?

---

## SCPI Command Reference

### CFP/CFP2

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:CFP:WDM:TYPE?

---

<b>Description</b>	<p>This query returns the WDM type of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE (10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:WDM:TYPE?
<b>Response Syntax</b>	<WDM_Type>
<b>Response(s)</b>	<p><b>WDM_Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the WDM Type of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:WDM:TYPE?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:CFP:CLEI:PRESeence?

---

---

**:SENSe[1..n]:DATA:TELeom:OPTical:CFP:CLEI:PRESe[nce]?**

---

<b>Description</b>	<p>This query returns the CLEI presence of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE (4 lanes)/100GE(10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:OPTical:CFP:CLEI:PRESe[nce]?
<b>Response Syntax</b>	<CLEI>
<b>Response(s)</b>	<p><b>CLEI:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the CLEI Presence of the CFP Connector module.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:CLEI:PRES?
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:OPTical:CFP:MODule:ID?

---

## SCPI Command Reference

### CFP/CFP2

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:CFP:MODE?

---

<b>Description</b>	<p>This query returns the mode value of the CFP Connector module.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 40GE (4 lanes)/100GE(4 lanes) /100GE(10 lanes) &gt; CFP</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:CFP:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the fiber mode of CFP.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:CFP:MODE?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:CFP:MODule:ID?

---

## Signal (Transport)

---

:SENSe[1..n]:DATA:TELeom:OPTical:TX:POWer?

---

<b>Description</b>	<p>This query returns the value of the transmitter power.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; TX Power (dBm)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:OPTical:TX:POWer? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the current status of an Optical alarm. The range for the Optical lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Power>
<b>Response(s)</b>	<p><b>Power:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of the transmitter power.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:TX:POW? 1
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:OPTical:RX:POWer?

---

## SCPI Command Reference

### Signal (Transport)

---

---

#### :SENSe[1..n]:DATA:TELeom:OPTical:RX:POWer?

---

<b>Description</b>	<p>This query returns the value of the receiver power.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; RX Power (dBm)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:OPTical:RX:POWer? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number value for which RX power is to be retrieved.</p> <p>Selects the lane for the current status of an Optical alarm. The range for the Optical lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Power>
<b>Response(s)</b>	<p><b>Power:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of the receiver power.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:RX:POW? 1
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:OPTical:TX:POWer?

---



---

**:SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer:MINimum?**

---

<b>Description</b>	<p>This query returns the value of optical lanes receiver power minimum.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; Min RX Power (dBm)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer:MINimum? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number value for which minimum RX power is to be retrieved.</p> <p>Selects the lane for the current status of an Optical alarm. The range for the Optical lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Minimum Power>
<b>Response(s)</b>	<p><b>Minimum Power:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of the minimum receiver power.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:RX:POW:MIN? 1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:TX:POWer?

---

## SCPI Command Reference

### Signal (Transport)

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer:MAXimum?

---

<b>Description</b>	<p>This query returns the value of optical lanes receiver power maximum.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; Max RX Power (dBm)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer:MAXimum? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number value for which maximum RX power is to be retrieved.</p> <p>Selects the lane for the current status of an Optical alarm. The range for the Optical lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Maximum Power>
<b>Response(s)</b>	<p><b>Maximum Power:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of the maximum receiver power.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:RX:POW:MAX? 1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer:MINimum?

---

---

**:SENSe[1..n]:DATA:TELeom:OPTical:WAVelength?**

---

<b>Description</b>	<p>This query returns the supported, the detected lane wavelength.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; Wavelength (nm)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:OPTical:WAVelength? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the current status of an Optical alarm. The range for the Optical lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Wavelength>
<b>Response(s)</b>	<p><b>Wavelength:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the supported, the detected lane wavelength.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:WAV? 1
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:OPTical:RX:POWer?

---

## SCPI Command Reference

### *Signal (Transport)*

---

**:OUTPut[1..n]:TELecom:LASer**

<b>Description</b>	<p>This command enables or disables the state of the laser.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; Laser ON/OFF &gt; All</p>
<b>Syntax</b>	<p>:OUTPut[1..n]:TELecom:LASer &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the laser.</p>
<b>Example(s)</b>	<p>OUTP:TEL:LAS ON</p> <p>OUTP:TEL:LAS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:ALASer</p>

---

---

**:OUTPut[1..n]:TELEcom:LASer?**

---

<b>Description</b>	<p>This query returns the current status of laser.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; Laser ON/OFF &gt; All</p>
<b>Syntax</b>	:OUTPut[1..n]:TELEcom:LASer?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the laser.</p> <p>1, laser is enabled.</p> <p>0, laser is disabled.</p>
<b>Example(s)</b>	<p>OUTP:TEL:LAS ON</p> <p>OUTP:TEL:LAS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>OUTPut[1..n]:TELEcom:LASer</p> <p>SENSe[1..n]:DATA:TELEcom:ALASer</p>

---

## SCPI Command Reference

### Signal (Transport)

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:POWer:RANGe?

---

<b>Description</b>	<p>This query returns the range between the minimum and maximum power range.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; Power Range (dBm)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:POWer:RANGe? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the current status of an Optical lane. The range for the Optical lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Power range>
<b>Response(s)</b>	<p><b>Power range:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the range between the minimum and maximum power range.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:POW:RANG? 1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:RX:POWer?

---

---

**:SENSe[1..n]:DATA:TELEcom:LOFF**

---

<b>Description</b>	<p>This command enables or disables the state of the laser.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE &gt; Physical Interface &gt; Laser OFF at Start up</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:LOFF <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the state of the Laser at startup.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:LOFF ON</p> <p>SENS:DATA:TEL:LOFF?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:LOFF</p> <p>SENSe[1..n]:DATA:TELEcom:LOFF?</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE</p>

---

## SCPI Command Reference

### *Signal (Transport)*

---

**:SENSe[1..n]:DATA:TELEcom:LOFF?**

<b>Description</b>	<p>This query returns the current status of the laser.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE &gt; Physical Interface &gt; Laser OFF at Start up</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:LOFF?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the laser Off at Startup.</p> <p>1, laser is enabled.</p> <p>0, laser is disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:LOFF ON</p> <p>SENS:DATA:TEL:LOFF?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:LOFF</p> <p>SENSe[1..n]:DATA:TELEcom:LOFF?</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE</p>

---



---

**:FETCh[1..n]:DATA:TELecom:LINK:GLOBal:STATus?**

---

<b>Description</b>	This query returns the status of Link. At *RST condition, this value is set to device-dependent. Navigation Path: Test Setup > Test Configurator > 100GE > LINK
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:LINK:GLOBal:STATus?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the link status.
<b>Example(s)</b>	FETC:DATA:TEL:LINK:GLOB:STAT?
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:ETH:ERR:PHYS:AUT?

---

## SCPI Command Reference

### *Signal (Transport)*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:FEC**

<b>Description</b>	<p>This command enables or disables the Forward Error Correction (FEC) for the Transmitter (TX) mode.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU &gt; Signal &gt; Signal Configuration &gt; OTU &gt; FEC</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:FEC &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Forward Error Correction (FEC) for the Transmitter (TX) mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:FEC ON</p> <p>SOUR:DATA:TEL:OTN:FEC?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:SCRambler</p>

---

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:FEC?**

---

<b>Description</b>	<p>This query returns the status of Forward Error Correction (FEC) for the Transmitter (TX) mode.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU &gt; Signal &gt; Signal Configuration &gt; OTU &gt; FEC</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:FEC?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Forward Error Correction (FEC) for the Transmitter (TX) mode.</p> <p>1, FEC for TX mode is enabled.</p> <p>0, FEC for TX mode is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:FEC ON</p> <p>SOUR:DATA:TEL:OTN:FEC?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:SCRambler?

---

## SCPI Command Reference

### *Signal (Transport)*

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler

---

<b>Description</b>	<p>This command enables or disables the scrambler for the Transmitter (TX) mode.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU &gt; Signal &gt; Signal Configuration &gt; OTU &gt; Scrambler</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the scrambler for the Transmitter (TX) mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU1:SCR ON</p> <p>SOUR:DATA:TEL:OTN:OTU1:SCR?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:FEC

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler?**

<b>Description</b>	<p>This query returns the status of scrambler for the Transmit (TX) mode.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU &gt; Signal &gt; Signal Configuration &gt; OTU &gt; Scrambler</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the scrambler for the Transmitter (TX) mode.</p> <p>1, scrambler for TX mode is enabled.</p> <p>0, scrambler for TX mode is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU1:SCR ON</p> <p>SOUR:DATA:TEL:OTN:OTU1:SCR?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:FEC?

---

## SCPI Command Reference

### *Signal (Transport)*

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler

---

**Description** This command enables or disables the scrambler for overclocked rates OTU3e1/2 of the transmitter.

At \*RST condition, this value is set to ON.

Navigation Path: Test > Setup > Test Configurator > OTU3e(1/2) > Signal > Signal Configuration > OTU3e(1/2) > Scrambler

**Syntax** :SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler <wsp>ON | OFF

**Parameter(s)** **Set:**  
The program data syntax for the first parameter is defined as a <Boolean Program Data> element.

The allowed elements for this parameter are: ON | OFF

Enables or disables the scrambler of the transmitter.

**Example(s)** SOUR:DATA:TEL:OTN:OTU3:E1:SCR ON

SOUR:DATA:TEL:OTN:OTU3:E1:SCR?

Returns: 1

**See Also** SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:FEC?

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler?**

<b>Description</b>	<p>This query returns the status of scrambler for overlocked rates OTU3e1/2 of the transmitter.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU3e(1/2) &gt; Signal &gt; Signal Configuration &gt; OTU3e(1/2) &gt; Scrambler</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the scrambler for the transmitter.</p> <p>1, returns the scrambler as ON.</p> <p>0, returns the scrambler as OFF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU3:E1:SCR ON</p> <p>SOUR:DATA:TEL:OTN:OTU3:E1:SCR?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:FEC

---

## SCPI Command Reference

### *Signal (Transport)*

---

**:SOURce[1..n]:DATA:TELecom:OTN:BTRaffic:PT[1..n]**

<b>Description</b>	<p>This command sets the background traffic for the instrument.</p> <p>At *RST condition, this value is set to PRBS31.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU &gt; Signal Configuration &gt; Background Traffic(Mux Type PT20/Mux Type PT21)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELecom:OTN:BTRaffic:PT[1..n] &lt;wsp&gt;AIS   NCLIENT   PRBS31   UNALlocated</p>
<b>Parameter(s)</b>	<p><b>Traffic:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   NCLIENT   PRBS31   UNALlocated</p> <p>Sets the value of the background traffic.</p> <p>AIS: AIS</p> <p>NCLIENT: NCLIENT</p> <p>PRBS31: PRBS31</p> <p>UNALlocated: Unallocated</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:BTR:PT21 PRBS31</p> <p>SOUR:DATA:TEL:OTN:BTR:PT21?</p> <p>Returns: PRBS31</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:E[1..n]:FEC?</p>



---

**:SOURce[1..n]:DATA:TELEcom:OTN:BTRaffic:PT[1..n]?**

---

<b>Description</b>	<p>This query returns the background traffic for the instrument.</p> <p>At *RST condition, this value is set to PRBS31.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU &gt; Signal Configuration &gt; Background Traffic(Mux Type PT21)</p> <p>Background traffic is present for OTN Multiplexing other than ""ODU3"" and ""ODU4"".</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:BTRaffic:PT[1..n]?
<b>Response Syntax</b>	<Traffic>
<b>Response(s)</b>	<p><b>Traffic:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the value of the background traffic.</p> <p>AIS, AIS is selected as background traffic.</p> <p>NCLIENT, NCLIENT is selected as background traffic.</p> <p>PRBS31, PRBS31 is selected as background traffic.</p> <p>UNALlocated, UNALlocated is selected as background traffic.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:BTR:PT20 PRBS31</p> <p>SOUR:DATA:TEL:OTN:BTR:PT20?</p> <p>Returns: PRBS31</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:FEC

---

## Frequency

---

**:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency?**

---

<b>Description</b>	<p>This query returns the frequency of the input signal for the transmitter of the optical port.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Frequency &gt; TX frequency</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency?
<b>Response Syntax</b>	<Frequency>
<b>Response(s)</b>	<p><b>Frequency:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the frequency of the input signal for the transmitter.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OPT:PORT:FREQ?
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency?</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue**

---

<b>Description</b>	<p>This command sets the offset value between the standard rate specification and the rate of input signal for the transmitter of the optical port.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Frequency &gt; Offset (PPM)</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue &lt;wsp&gt; &lt;Value&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Value:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the offset value between the standard rate specification and the rate of input signal for the transmitter of the optical port.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL 15 SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue? SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency?</pre>

---

## SCPI Command Reference

### Frequency

---

**:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue?**

<b>Description</b>	<p>This query returns the offset value between the standard rate specification and the rate of input signal for the transmitter of the optical port.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Frequency &gt; Offset (PPM)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the offset value between the standard rate specification and the rate of input signal for the transmitter of the optical port.</p> <p>This parameter is optional. If no token is specified, the current frequency offset value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Offset&gt;</p>
<b>Response(s)</b>	<p><b>Offset:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the offset value between the standard rate specification and the rate of input signal.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL 15</p> <p>SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue</p>

---

**:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet**

---

<b>Description</b>	<p>This command enables or disables the frequency offset generation of optical port.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; TX Frequency &gt; Offset &gt; ON/OFF</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the frequency offset generation.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL 15 SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS ON SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS? Returns: 1</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue?</code>

---

## SCPI Command Reference

### Frequency

---

---

#### :SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet?

---

<b>Description</b>	<p>This query returns the frequency offset generation of the optical port.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; TX Frequency &gt; Offset &gt; ON/OFF</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet?
<b>Response Syntax</b>	<Offset>
<b>Response(s)</b>	<p><b>Offset:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frequency offset generation of an optical port.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL 15</p> <p>SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS ON</p> <p>SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue

---

---

**:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency?**

<b>Description</b>	<p>This query returns the frequency of the input signal for the receiver of the optical port.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Frequency &gt; RX Frequency &gt; Frequency (GHz)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the Received frequency. The range for the lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Frequency>
<b>Response(s)</b>	<p><b>Frequency:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the frequency of the input signal for the receiver.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:PORT:FREQ? 0
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:POSitive?

---

## SCPI Command Reference

### Frequency

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:POSitive?

---

<b>Description</b>	<p>This query returns the offset between the standard rate specification and the largest rate recorded from the received signal of the optical port.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Frequency &gt; RX Frequency &gt; Max Positive Offset (PPM)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:POSitive? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the Received Positive frequency. The range for the lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Poffset>
<b>Response(s)</b>	<p><b>Poffset:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the positive frequency of the input signal.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:PORT:FREQ:POS? 0
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:NEGative?

---



**:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:NEGative?**

<b>Description</b>	<p>This query returns the offset between the standard rate specification and the smallest rate recorded from the received signal of the optical port.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Frequency &gt; RX Frequency &gt; Max Negative Offset (PPM)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:NEGative? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the Received Negative frequency. The range for the lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Noffset>
<b>Response(s)</b>	<p><b>Noffset:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the negative frequency of the input signal.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:PORT:FREQ:NEG? 0
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:POSitive?

## SCPI Command Reference

### Frequency

---

---

#### :SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue?

---

<b>Description</b>	<p>This query returns the offset value between the standard rate specification and the rate of input signal for the receiver of the optical port.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Frequency &gt; RX Frequency &gt; Offset (PPM)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OPTical:PORT:FREQuency:OFFSet:VALue? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the Received frequency offset. The range for the lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Offset>
<b>Response(s)</b>	<p><b>Offset:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frequency offset value for the receiver.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OPTical:TX:POWer?

---

## Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16**

---

<b>Description</b>	<p>This command sets the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16 <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU1:SM:SAPI:B16 "EXFO OTU SAPI"</p> <p>SOUR:DATA:TEL:OTN:OTU1:SM:SAPI:B16?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16?

---

<b>Description</b>	<p>This query returns the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU1:SM:SAPI:B16 "EXFO OTU SAPI"</p> <p>SOUR:DATA:TEL:OTN:OTU1:SM:SAPI:B16?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>Note(s)</b>	<STRING RESPONSE DATA>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16**

---

<b>Description</b>	<p>This command sets the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16 <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU1:SM:DAPI:B16 "EXFO OTU DAPI"</p> <p>SOUR:DATA:TEL:OTN:OTU1:SM:DAPI:B16?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:SM:DAPI:B16?**

<b>Description</b>	<p>This query returns the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:SM:DAPI:B16?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU1:SM:DAPI:B16 "EXFO OTU DAPI"</p> <p>SOUR:DATA:TEL:OTN:OTU1:SM:DAPI:B16?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>Note(s)</b>	<STRING RESPONSE DATA>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:SM:SAPI:B16</p> <p>SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:SM:SAPI:B16?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B32**

---

<b>Description</b>	<p>This command sets the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B32 <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU1:SM:OPSP:B32 "EXFO OTU OPERATOR SPECIFIC"</p> <p>SOUR:DATA:TEL:OTN:OTU1:SM:OPSP:B32?</p> <p>Returns: "EXFO OTU OPERATOR SPECIFIC"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B32?**

<b>Description</b>	<p>This query returns the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B32?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU1:SM:OPSP:B32 "EXFO OTU OPERATOR SPECIFIC"</p> <p>SOUR:DATA:TEL:OTN:OTU1:SM:OPSP:B32?</p> <p>Returns: "EXFO OTU OPERATOR SPECIFIC"</p>
<b>Note(s)</b>	<STRING RESPONSE DATA>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16?</p>

---



---

**:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B**

---

<b>Description</b>	<p>This command sets the injected message for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:SAPI:B "EXFO OTU SAPI"</p> <p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:SAPI:B?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B?

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B?**

<b>Description</b>	<p>This query returns the injected message for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:SAPI:B "EXFO OTU SAPI"</p> <p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:SAPI:B?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B

---

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B**

---

<b>Description</b>	<p>This command sets the injected message for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; OTU3e(1/2)&gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:DAPI:B "EXFO OTU DAPI"</p> <p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:DAPI:B?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B?

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B?**

<b>Description</b>	<p>This query returns the injected message for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2)&gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:DAPI:B "EXFO OTU DAPI"</p> <p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:DAPI:B?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:OPSPec:B**

---

<b>Description</b>	<p>This command sets the injected message for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:OPSPec:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:OPSP:B "EXFO OTU OPERATOR SPECIFIC"</p> <p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:OPSP:B?</p> <p>Returns: "EXFO OTU OPERATOR SPECIFIC"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B?

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:OPSPec:B?**

<b>Description</b>	<p>This query returns the injected message for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:OPSP:B"EXFO OTU OPERATOR SPECIFIC"</p> <p>SOUR:DATA:TEL:OTN:OTU3:E1:SM:OPSP:B?</p> <p>Returns: "EXFO OTU OPERATOR SPECIFIC"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B

---

---

**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:SAPI:EXPEcted**

---

<b>Description</b>	<p>This command sets the expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:SAPI:EXPEcted <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for the OTU instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU1:TTI:SAPI:EXP "EXFO OTU SAPI"</p> <p>SENS:DATA:TEL:OTN:OTU1:TTI:SAPI:EXP?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:TIM</p> <p>SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n][1..n]:TTI:SAPI:EXPEcted?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

---

#### :SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:EXpected?

---

<b>Description</b>	<p>This query returns the expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:EXpected?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for the OTU.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU1:TTI:SAPI:EXP "EXFO OTU SAPI"</p> <p>SENS:DATA:TEL:OTN:OTU1:TTI:SAPI:EXP?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXpected</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXpected?</p>

---



---

**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:DAPI:EXPEcted**

<b>Description</b>	<p>This command sets expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:DAPI:EXPEcted <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for the OTU instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU1:TTI:TIM DAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU1:TTI:DAPI:EXP "EXFO OTU DAPI"</p> <p>SENS:DATA:TEL:OTN:OTU1:TTI:DAPI:EXP?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:SAPI:EXPEcted</p> <p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:SAPI:EXPEcted?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

---

#### :SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:EXPeCted?

---

<b>Description</b>	<p>This query returns expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for the OTU.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU1:TTI:TIM DAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU1:TTI:DAPI:EXP "EXFO OTU DAPI"</p> <p>SENS:DATA:TEL:OTN:OTU1:TTI:DAPI:EXP?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPeCted</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPeCted?</p>

---

**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPEcted**

<b>Description</b>	<p>This command sets the expected message for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPEcted <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:SAPI:EXP "EXFO OTU SAPI"</p> <p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:SAPI:EXP?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:E[1..n]:TTI:TIM

## SCPI Command Reference

### Traces (OTN)

---

---

#### :SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPeCted?

---

<b>Description</b>	<p>This query returns the expected message for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:SAPI:EXP "EXFO OTU SAPI"</p> <p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:SAPI:EXP?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPEcted**

<b>Description</b>	<p>This command sets expected message for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2)&gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPEcted <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:TIM DAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:DAPI:EXP "EXFO OTU DAPI"</p> <p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:DAPI:EXP?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM

## SCPI Command Reference

### Traces (OTN)

---

---

#### :SENSe[1..n]:DATA:TELeom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPeCted?

---

<b>Description</b>	<p>This query returns expected message for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:TIM DAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:DAPI:EXP "EXFO OTU DAPI"</p> <p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:DAPI:EXP?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:OTN:OTU[1..n]:E[1..n]:TTI:TIM

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM**

<b>Description</b>	<p>This command enables or disables the state of Trace Identifier Mismatch (TIM) for the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM <wsp>SAPI   DAPI, ON   OFF
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI: the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU3:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU3:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM?</p>

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM?**

<b>Description</b>	<p>This query returns state of Trace Identifier Mismatch (TIM) for the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM? <wsp>SAPI   DAPI
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI: the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the status of Trace Identifier Mismatch (TIM).</p> <p>SAPI, indicates the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, indicates the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU3:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU3:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM?</p>



**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:E[1..n]:TTI:TIM**

<b>Description</b>	<p>This command enables or disables the state of Trace Identifier Mismatch (TIM) for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:E[1..n]:TTI:TIM <wsp>SAPI   DAPI, ON   OFF
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p> <p>SAPI: the SAPI which allows editing of the Source Access Point Identifier (SAPI) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the Destination Access Point Identifier (DAPI) message to be generated.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:E[1..n]:TTI:TIM?

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM?**

<b>Description</b>	<p>This query returns status of Trace Identifier Mismatch (TIM) for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM? <wsp>SAPI   DAPI
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p> <p>SAPI: the SAPI which allows editing of the Source Access Point Identifier (SAPI) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the Destination Access Point Identifier (DAPI) message to be generated.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Trace Identifier Mismatch (TIM).</p> <p>0, status of the Trace Identifier Mismatch (TIM) is disabled</p> <p>1, status of the Trace Identifier Mismatch (TIM) is enabled</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:OTU3:E1:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:SAPI:B**

---

<b>Description</b>	<p>This command sets the generated message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to ""EXFO ODU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:SAPI:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Is the generated message for ODU TTI Trace</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:PM:SAPI:B "EXFO ODU SAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:PM:SAPI:B?</p> <p>Returns: "EXFO ODU SAPI"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:DAPI:B</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:DAPI:B?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:PM:SAPI:B?**

<b>Description</b>	<p>This query returns the generated message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to ""EXFO ODU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:PM:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:PM:SAPI:B "EXFO ODU SAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:PM:SAPI:B?</p> <p>Returns: "EXFO ODU SAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:SM:SAPI:B

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:DAPI:B**

<b>Description</b>	<p>This command sets the generated message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to ""EXFO ODU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:DAPI:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Is the generated message for ODU TTI Trace</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:PM:DAPI:B "EXFO ODU DAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:PM:DAPI:B?</p> <p>Returns: "EXFO ODU DAPI"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:SAPI:B</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:SAPI:B?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:PM:DAPI:B?**

<b>Description</b>	<p>This query returns the generated message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to ""EXFO ODU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:PM:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:PM:DAPI:B "EXFO ODU DAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:PM:DAPI:B?</p> <p>Returns: "EXFO ODU DAPI"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:PM:SAPI:B</p> <p>SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:PM:SAPI:B?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:OPSPec:B**

<b>Description</b>	<p>This command sets the generated message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to ""EXFO ODU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:OPSPec:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Is the generated message for ODU TTI Trace</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:PM:OPSP:B "EXFO ODU OPERATOR SPECIFIC"</p> <p>SOUR:DATA:TEL:OTN:ODU3:PM:OPSP:B?</p> <p>Returns: "EXFO ODU OPERATOR SPECIFIC"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:OPSPec:B?**

<b>Description</b>	<p>This query returns the generated message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to ""EXFO ODU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:PM:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:PM:OPSP:B "EXFO ODU OPERATOR SPECIFIC"</p> <p>SOUR:DATA:TEL:OTN:ODU3:PM:OPSP:B?</p> <p>Returns: "EXFO ODU OPERATOR SPECIFIC"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B?</p>

---



**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:SAPI:B**

<b>Description</b>	<p>This command sets the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:SAPI:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:SAPI:B "EXFO OTU SAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:SAPI:B?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B?

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:SAPI:B?**

<b>Description</b>	<p>This query returns the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:SAPI:B "EXFO OTU SAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:SAPI:B?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:SAPI:B

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:DAPI:B**

<b>Description</b>	<p>This command sets the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:DAPI:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:DAPI:B "EXFO OTU DAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:DAPI:B?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B?

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:DAPI:B?**

<b>Description</b>	<p>This query returns the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:DAPI:B "EXFO OTU DAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:DAPI:B?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:DAPI:B

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:OPSPec:B**

<b>Description</b>	<p>This command sets the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:OPSPec:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:OPSP:B "EXFO OTU OPERATOR SPECIFIC"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:OPSP:B?</p> <p>Returns: "EXFO OTU OPERATOR SPECIFIC"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:OPSPec:B?

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:OPSPec:B?**

<b>Description</b>	<p>This query returns the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:PM:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:OPSP:B "EXFO OTU OPERATOR SPECIFIC"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:PM:OPSP:B?</p> <p>Returns: "EXFO OTU OPERATOR SPECIFIC"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SM:OPSPec:B

**:SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:SAPI:EXPEcted**

<b>Description</b>	<p>This command sets the expected message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to ""EXFO ODU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>In Multi-Channel OTN, when accessing an ODU LO, the channel must be set using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN. The expected message can also be set for all channels by using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:ACHA.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:SAPI:EXPEcted <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Is the expected message for ODU TTI Trace</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:OTN:ODU3:TTI:TIM SAPI,ON SENS:DATA:TEL:OTN:ODU3:TTI:SAPI:EXP "EXFO ODU SAPI" SENS:DATA:TEL:OTN:ODU3:TTI:SAPI:EXP? Returns: "EXFO ODU SAPI"  For Multi-Channel OTN: SENS:DATA:TEL:OTN:ODU100:TTI:ACHA ON SENS:DATA:TEL:OTN:ODU100:TTI:TIM SAPI,ON SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP "XFO ODU SAPI" SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP? Returns: "XFO ODU SAPI"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:DAPI:EXPEcted SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:DAPI:EXPEcted? SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:CHAN SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:ACHA</pre>

## SCPI Command Reference

### Traces (OTN)

---

---

#### :SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:SAPI:EXPeCted?

---

<b>Description</b>	<p>This query returns the expected message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to ""EXFO ODU SAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>In Multi-Channel OTN, when accessing an ODU LO, the channel must be set using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN. The expected message can also be set for all channels by using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:ACHA.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:SAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:TTI:SAPI:EXP "EXFO ODU SAPI"</p> <p>SENS:DATA:TEL:OTN:ODU3:TTI:SAPI:EXP?</p> <p>Returns: "EXFO ODU SAPI"</p> <p>For Multi-Channel OTN:</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:ACHA ON</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP "XFO ODU SAPI"</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP?</p> <p>Returns: "XFO ODU SAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:DAPI:EXPeCted</p> <p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:DAPI:EXPeCted?</p> <p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:CHAN</p> <p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:ACHA</p>

---



**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXPEcted**

<b>Description</b>	<p>This command sets the expected message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to ""EXFO ODU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>In Multi-Channel OTN, when accessing an ODU LO, the channel must be set using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN. The expected message can also be set for all channels by using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:ACHA.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXPEcted <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Is the expected message for ODU TTI Trace</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:OTN:ODU3:TTI:TIM DAPI,ON SENS:DATA:TEL:OTN:ODU3:TTI:DAPI:EXP "EXFO ODU DAPI" SENS:DATA:TEL:OTN:ODU3:TTI:DAPI:EXP? Returns: "EXFO ODU DAPI"  For Multi-Channel OTN: SENS:DATA:TEL:OTN:ODU100:TTI:ACHA ON SENS:DATA:TEL:OTN:ODU100:TTI:TIM DAPI,ON SENS:DATA:TEL:OTN:ODU100:TTI:DAPI:EXP "XFO ODU DAPI" SENS:DATA:TEL:OTN:ODU100:TTI:DAPI:EXP? Returns: "XFO ODU DAPI"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPEcted SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPEcted? SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHAN SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHA</pre>

## SCPI Command Reference

### Traces (OTN)

---

---

#### :SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:DAPI:EXPeCted?

---

<b>Description</b>	<p>This query returns the expected message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to ""EXFO ODU DAPI"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>In Multi-Channel OTN, when accessing an ODU LO, the channel must be set using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN. The expected message can also be set for all channels by using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:ACHA.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:DAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:TTI:TIM DAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:TTI:DAPI:EXP "EXFO ODU DAPI"</p> <p>SENS:DATA:TEL:OTN:ODU3:TTI:DAPI:EXP?</p> <p>Returns: "EXFO ODU DAPI"</p> <p>For Multi-Channel OTN:</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:ACHA ON</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:TIM DAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:DAPI:EXP "XFO ODU DAPI"</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:DAPI:EXP?</p> <p>Returns: "XFO ODU DAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:SAPI:EXPeCted</p> <p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:SAPI:EXPeCted?</p> <p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:CHAN</p> <p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:ACHA</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:EXPEcted**

---

<b>Description</b>	<p>This command sets the expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO ODU SAPI".</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU(1/2)e</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:EXPEcted &lt;wsp&gt; &lt;Message&gt;</p>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:SAPI:EXP "EXFO ODU SAPI"</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:SAPI:EXP?</p> <p>Returns: "EXFO ODU SAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPEcted?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:EXPeCted?**

<b>Description</b>	<p>This query returns the expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO ODU SAPI"".</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt;ODU(1/2)e</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:SAPI:EXP "EXFO ODU SAPI"</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:SAPI:EXP?</p> <p>Returns: "EXFO ODU SAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:EXPeCted</p>

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:EXpected**

<b>Description</b>	<p>This command sets expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO ODU DAPI"".</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:EXpected <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:TIM DAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:DAPI:EXP "EXFO ODU DAPI"</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:DAPI:EXP?</p> <p>Returns: "EXFO ODU DAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXpected?</p>

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:EXPeCted?**

<b>Description</b>	<p>This query returns expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO ODU DAPI"".</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:TIM DAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:DAPI:EXP "EXFO ODU DAPI"</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:DAPI:EXP?</p> <p>Returns: "EXFO ODU DAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:TIM</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:EXPeCted</p>

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM**

<b>Description</b>	<p>This command enables or disables the state of Trace Identifier Mismatch (TIM) for the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; TTI Traces &gt; ODU</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM <wsp>SAPI   DAPI, ON   OFF
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI: the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM?</p>

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:TIM?**

<b>Description</b>	<p>This query returns state of Trace Identifier Mismatch (TIM) for the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:TIM? &lt;wsp&gt;SAPI   DAPI</code>
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI: the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
<b>Response Syntax</b>	<code>&lt;Set&gt;</code>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the status of Trace Identifier Mismatch (TIM).</p> <p>ON, enables the Trace Identifier Mismatch (TIM).</p> <p>OFF, disables the Trace Identifier Mismatch (TIM).</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:OTN:ODU3:TTI:TIM SAPI,ON SENS:DATA:TEL:OTN:ODU3:TTI:TIM? SAPI Returns: 1</pre>
<b>Note(s)</b>	<code>&lt;NR1 NUMERIC RESPONSE DATA&gt;</code>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:TIM SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:TIM?</pre>

---



**:SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:E[1..n]:TTI:TIM**

<b>Description</b>	<p>This command enables or disables the state of Trace Identifier Mismatch (TIM) for the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU(1/2)e</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:E[1..n]:TTI:TIM <wsp>SAPI   DAPI, ON   OFF
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI: the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Trace Identifier Mismatch (TIM)</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:OTN:OTU[1..n]:TTI:TIM?

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:TIM?**

<b>Description</b>	<p>This query returns state of Trace Identifier Mismatch (TIM) for the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results &gt; TTI Traces &gt; ODU(1/2)e</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:TIM? &lt;wsp&gt;SAPI   DAPI</p>
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI: the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
<b>Response Syntax</b>	<p>&lt;Set&gt;</p>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Trace Identifier Mismatch (TIM).</p> <p>ON, enables the Trace Identifier Mismatch (TIM).</p> <p>OFF, disables the Trace Identifier Mismatch (TIM).</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM</p>

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:GOVEverwrite?**

<b>Description</b>	<p>This query returns if an expected SAPI ODU TTI Traces has been overwritten on at least one channel.</p> <p>Navigation Path: Test &gt; Setup &gt; ODU Channels &gt; Traces &gt; PM TTI Traces</p> <p>NOTE: This query is only available in Multi-Channel OTN.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:GOVEverwrite?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>1: At least one expected SAPI ODU TTI Trace has been overwritten.</p> <p>0: All expected SAPI ODU TTI Traces are the same as the Global expected ODU TTI Trace.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:GOVE?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXpected SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHAN SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHA

---

## SCPI Command Reference

### Traces (OTN)

---

---

#### **:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:GOVErwrite?**

---

<b>Description</b>	<p>This query returns if an expected DAPI ODU TTI Traces has been overwritten on at least one channel.</p> <p>Navigation Path: Test &gt; Setup &gt; ODU Channels &gt; Traces &gt; PM TTI Traces</p> <p>NOTE: This query is only available in Multi-Channel OTN.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:GOVErwrite?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>1: At least one expected DAPI ODU TTI Trace has been overwritten.</p> <p>0: All expected DAPI ODU TTI Traces are the same as the Global expected ODU TTI Trace.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU100:TTI:DAPI:GOVE?
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXPeCted</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHAN</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHA</p>

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHannel**

<b>Description</b>	<p>This command forces the ODU TTI Traces commands/queries to be applied on all channels (ON) or a single channel (OFF).</p> <p>At *RST condition, this value is set to OFF.</p> <p>NOTE: This command is only available in Multi-Channel OTN.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHannel <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>AllChannel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Selects if the ODU TTI Traces commands/queries are applied on all channels (ON) or a single channel (OFF).</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:OTN:ODU100:TTI:ACHA ON SENS:DATA:TEL:OTN:ODU100:TTI:TIM SAPI,ON SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP "XFO ODU SAPI" SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP? Returns: "XFO ODU SAPI"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHAN SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHA?</pre>

## SCPI Command Reference

### Traces (OTN)

---

---

#### :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHannel?

---

<b>Description</b>	<p>This query returns if the ODU TTI Traces commands/queries are applied on all channels (ON) or a single channel (OFF).</p> <p>At *RST condition, this value is set to OFF.</p> <p>NOTE: This command is only available in Multi-Channel OTN.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHannel?
<b>Response Syntax</b>	<AllChannel>
<b>Response(s)</b>	<p><b>AllChannel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns an indication that the ODU TTI Traces commands/queries are to be applied on all channels.</p> <p>1, ODU TTI Trace commands/queries are applied on all channels.</p> <p>0, ODU TTI Trace commands/queries are applied on a specific channels determined by SENS:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU100:TTI:ACHA ON</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP "XFO ODU SAPI"</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP?</p> <p>Returns: "XFO ODU SAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHAN</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHA</p>

---

---

**:SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:CHANnel**

<b>Description</b>	<p>This command sets the value of the channel used by the ODU TTI Traces.</p> <p>At *RST condition, this value is set to 1.</p> <p>When setting a channel value, the AChannel value will automatically be set to OFF.</p> <p>NOTE: This command is only available in Multi-Channel OTN.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:CHANnel <wsp> <Channel>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel used for ODU TTI Traces, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:OTN:ODU100:TTI:CHAN 3 SENS:DATA:TEL:OTN:ODU100:TTI:TIM SAPI,ON SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP "XFO ODU SAPI3" SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP? Returns: "XFO ODU SAPI3"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:CHAN? SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:ACHA</pre>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHANnel?**

<b>Description</b>	<p>This query returns the value of the channel used by the ODU TTI Traces.</p> <p>At *RST condition, this value is set to 1.</p> <p>When setting a channel value, the AChannel value will automatically be set to OFF.</p> <p>NOTE: This command is only available in Multi-Channel OTN.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHANnel?
<b>Response Syntax</b>	<Channel>
<b>Response(s)</b>	<p><b>Channel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the channel used for the ODU TTI Trace, when AllCHAnnel is 'OFF', in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU100:TTI:CHAN 3</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP "XFO ODU SAPI3"</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:SAPI:EXP?</p> <p>Returns: "XFO ODU SAPI3"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:CHAN</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:ACHA</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B**

<b>Description</b>	<p>This command sets the generated message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO TCM SAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU4:TCM1:SAPI:B "EXFO"</p> <p>SOUR:DATA:TEL:OTN:ODU4:TCM1:SAPI:B?</p> <p>Returns: "EXFO"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B?**

<b>Description</b>	<p>This query returns the generated message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO TCM SAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU4:TCM1:SAPI:B "EXFO"</p> <p>SOUR:DATA:TEL:OTN:ODU4:TCM1:SAPI:B?</p> <p>Returns: "EXFO"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B**

<b>Description</b>	<p>This command sets the generated message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO TCM DAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU4:TCM1:DAPI:B "EXFO1"</p> <p>SOUR:DATA:TEL:OTN:ODU4:TCM1:DAPI:B?</p> <p>Returns: "EXFO1"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B?**

<b>Description</b>	<p>This query returns the generated message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO TCM DAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU4:TCM1:DAPI:B "EXFO1"</p> <p>SOUR:DATA:TEL:OTN:ODU4:TCM1:DAPI:B?</p> <p>Returns: "EXFO1"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:OPSPec:B**

---

<b>Description</b>	<p>This command sets the generated message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO TCM OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; Operator Specific</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:OPSPec:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU4:TCM1:OPSP:B "EXFO INDIA "</p> <p>SOUR:DATA:TEL:OTN:ODU4:TCM1:OPSP:B?</p> <p>Returns: "EXFO INDIA "</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B?</p>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:OPSPec:B?**

<b>Description</b>	<p>This query returns the generated message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO TCM OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; Operator Specific</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU4:TCM1:OPSP:B "EXFO INDIA "</p> <p>SOUR:DATA:TEL:OTN:ODU4:TCM1:OPSP:B?</p> <p>Returns: "EXFO INDIA "</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:SAPI:B?</p>

---

---

<b>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:SAPI:B</b>	
<b>Description</b>	<p>This command sets the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI""</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OUT3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2) -TCM &gt; TCM TTI Traces &gt; SAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; ODU(1/2)e -TCM &gt; TCM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:SAPI:B <wsp><Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:SAPI:B "EXFO OTU SAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:SAPI:B?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16?

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:SAPI:B?**

<b>Description</b>	<p>This query returns the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI""</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; SAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; ODU(1/2)e -TCM &gt; TCM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:SAPI:B "EXFO OTU SAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:SAPI:B?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:SAPI:B16



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:DAPI:B**

---

<b>Description</b>	<p>This command sets the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; DAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; ODU(1/2)e -TCM &gt; TCM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:DAPI:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:DAPI:B "EXFO OTU DAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:DAPI:B?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16?

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:DAPI:B?**

<b>Description</b>	<p>This query returns the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; DAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; ODU(1/2)e -TCM &gt; TCM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:DAPI:B "EXFO OTU DAPI"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:DAPI:B?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:DAPI:B16

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:OPSPec:B**

<b>Description</b>	<p>This command sets the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; Operator Specific</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; ODU(1/2)e -TCM &gt; TCM TTI Traces &gt; Operator Specific</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:OPSPec:B <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:OPSP:B "EXFO OTU OPERATOR SPECIFIC"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:OPSP:B?</p> <p>Returns: "EXFO OTU OPERATOR SPECIFIC"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B32?

---

## SCPI Command Reference

### Traces (OTN)

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:OPSPec:B?**

<b>Description</b>	<p>This query returns the injected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; Operator Specific</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup &gt; OTU4 &gt; TTI Traces &gt; ODU1/2e &gt; ODU1/2e -TCM &gt; TCM TTI Traces &gt; Operator Specific</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:OPSP:B "EXFO OTU OPERATOR SPECIFIC"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:TCM1:OPSP:B?</p> <p>Returns: "EXFO OTU OPERATOR SPECIFIC"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SM:OPSPec:B32

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPeCted**

<b>Description</b>	<p>This command sets the expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO TCM SAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; SAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPeCted <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:SAPI:EXP "EXFO TCM1 SAPI"</p> <p>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:SAPI:EXP?</p> <p>Returns: "EXFO TCM1 SAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted?</p>

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPeCted?**

<b>Description</b>	<p>This query returns the expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO TCM SAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; SAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:SAPI:EXP "EXFO TCM1 SAPI"</p> <p>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:SAPI:EXP?</p> <p>Returns: "EXFO TCM1 SAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted</p> <p>SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted**

---

<b>Description</b>	<p>This command sets expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO TCM DAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; DAPI</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted &lt;wsp&gt; &lt;Message&gt;</pre>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for the instrument.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:DAPI:EXP "EXFO TCM1 DAPI" SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:DAPI:EXP? Returns: "EXFO TCM1 DAPI"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPeCted SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPeCted?</pre>

---

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted?**

<b>Description</b>	<p>This query returns expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO TCM DAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; DAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:DAPI:EXP "EXFO TCM1 DAPI"</p> <p>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:DAPI:EXP?</p> <p>Returns: "EXFO TCM1 DAPI"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPeCted</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:EXPeCted?</p>

---



---

```
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted
```

---

<b>Description</b>	<p>This command sets the expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; SAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU4 &gt; TTI Traces &gt; ODU2e(1/2) &gt; ODU2e(1/2)-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; SAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:EXPEcted <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for the instrument.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:TIM SAPI,ON SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:SAPI:EXP "EXFO OTU SAPI" SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:SAPI:EXP? Returns: "EXFO OTU SAPI"</pre>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:EXPEcted?

---

## SCPI Command Reference

### Traces (OTN)

---

<b>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:EXPeCted?</b>	
<b>Description</b>	<p>This query returns the expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; SAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU4 &gt; TTI Traces &gt; ODU2e(1/2) &gt; ODU2e(1/2)-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; SAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:SAPI:EXP "EXFO OTU SAPI"</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:SAPI:EXP?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:EXPeCted

---

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted**

---

<b>Description</b>	<p>This command sets expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; DAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU4 &gt; TTI Traces &gt; ODU2e(1/2) &gt; ODU2e(1/2)-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; DAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:TIM DAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:DAPI:EXP "EXFO OTU DAPI"</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:DAPI:EXP?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:EXPeCted?

---

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted?**

<b>Description</b>	<p>This query returns expected message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; DAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU4 &gt; TTI Traces &gt; ODU2e(1/2) &gt; ODU2e(1/2)-TCM &gt; TCM TTI Traces &gt; Expected Messages &gt; DAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for the instrument.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:TIM DAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:DAPI:EXP "EXFO OTU DAPI"</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:DAPI:EXP?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:EXPeCted

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:TIM**

<b>Description</b>	<p>This command enables or disables the state of Trace Identifier Mismatch (TIM) for ODU TCM.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; TCP-TIM</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:TIM <wsp>SAPI   DAPI, ON   OFF
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for ODU TCM TTI.</p> <p>SAPI: the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>Note(s)</b>	SENSe:DATA:TELEcom:OTN:OTU:TTI:SAPI:EXpected
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIMer</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIMer?</p>

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:TIM?**

<b>Description</b>	<p>This query returns state of Trace Identifier Mismatch (TIM) for ODU TCM TTI.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; TCP-TIM</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:TIM? &lt;wsp&gt;SAPI   DAPI</p>
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Sets the expected message for the instrument.</p> <p>SAPI: the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
<b>Response Syntax</b>	<p>&lt;Set&gt;</p>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Trace Identifier Mismatch (TIM).</p> <p>1, enables the status of TIM.</p> <p>0, disables the status of TIM.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU4:TCM1:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>Note(s)</b>	<p>SENSe:DATA:TELEcom:OTN:OTU:TTI:SAPI:EXPeCted?</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIMer</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIMer?</p>

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:TIM**

<b>Description</b>	<p>This command enables or disables the state of Trace Identifier Mismatch (TIM) for the instrument.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; TCP-TIM</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU4 &gt; TTI Traces &gt; ODU(1/2)e &gt; ODU(1/2)e-TCM &gt; TCM TTI Traces &gt; TCP-TIM</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:TIM &lt;wsp&gt;SAPI   DAPI, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI: the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Trace Identifier Mismatch (TIM)</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:TIM?</p>

## SCPI Command Reference

### Traces (OTN)

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:TIM?**

<b>Description</b>	<p>This query returns state of Trace Identifier Mismatch (TIM) for the instrument.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU3e(1/2) &gt; TTI Traces &gt; ODU3e(1/2) &gt; ODU3e(1/2)-TCM &gt; TCM TTI Traces &gt; TCP-TIM</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Setup/Results &gt; OTU4 &gt; TTI Traces &gt; ODU1/2e &gt; ODU1/2e-TCM &gt; TCM TTI Traces &gt; TCP-TIM</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:TIM? &lt;wsp&gt;SAPI   DAPI</p>
<b>Parameter(s)</b>	<p><b>Etim:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAPI   DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI: the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI: the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
<b>Response Syntax</b>	<p>&lt;Set&gt;</p>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Trace Identifier Mismatch (TIM).</p> <p>ON, enables the Trace Identifier Mismatch (TIM).</p> <p>OFF, disables the Trace Identifier Mismatch (TIM).</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:TIM SAPI,ON</p> <p>SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:TIM? SAPI</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM</p>



## FTFL/PT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication**

---

**Description** This command sets the selection of the Fault Type Fault Location (FTFL) Fault Indication message to be generated.

At \*RST condition, this value is set to NFAult.

Navigation Path: Test > Setup > OTN BERT > OTU > FTFL/PT > ODU > FTFL

**Syntax** :SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication <wsp>FORWARD | BACKward, NFAult | SFAil | SDEGrade | REServed

---

## SCPI Command Reference

### FTFL/PT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication**

<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWard   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p> <p><b>Indication:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NFAult   SFAil   SDEGrade   REServed</p> <p>Sets the fault indication of the Fault Type Fault Location (FTFL) fault indicator message to be generated.</p> <p>NFAult: NFAult (No Fault)</p> <p>SFAil: SFAil (Signal Fail)</p> <p>SDEGrade: SDEGrade (Signal Degraded)</p> <p>REServed: Reserved</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU1:FTFL:IND FORW,SFA</p> <p>SOUR:DATA:TEL:OTN:ODU1:FTFL:IND? FORW</p> <p>Returns: SFAIL</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier?</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?**

<b>Description</b>	<p>This query returns the selection of the Fault Type Fault Location (FTFL) Fault Indication message to be generated.</p> <p>At *RST condition, this value is set to NFAult.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; FTFL</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication? <wsp>FORWARD   BACKward
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Indication>
<b>Response(s)</b>	<p><b>Indication:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Fault Type Fault Location (FTFL) Fault Indication message to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU1:FTFL:IND FORW,SFA</p> <p>SOUR:DATA:TEL:OTN:ODU1:FTFL:IND? FORW</p> <p>Returns: SFAIL</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENTifier</p> <p>SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENTifier?</p>

## SCPI Command Reference

### FTFL/PT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE**

<b>Description</b>	<p>This command sets the selection of the Fault Type Fault Location (FTFL) Fault Indication code to be generated.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; FTFL</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE &lt;wsp&gt;FORWard   BACKward, &lt;Code&gt;</p>
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWard   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p> <p><b>Code:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the selection of the Fault Type Fault Location (FTFL) fault indicator code to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU1:FTFL:CODE FORW,#H01</p> <p>SOUR:DATA:TEL:OTN:ODU1:FTFL:CODE? FORW</p> <p>Returns: 1 (the decimal form of "#01H")</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication</p> <p>LINS10:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE?**

<b>Description</b>	<p>This query returns the selection of the Fault Type Fault Location (FTFL) Fault Indication code to be generated.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; FTFL</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE? <wsp>FORWard   BACKward
<b>Parameter(s)</b>	<p><b>File:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWard   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the selection of the Fault Type Fault Location (FTFL) fault indicator code to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU1:FTFL:CODE FORW,#H01</p> <p>SOUR:DATA:TEL:OTN:ODU1:FTFL:CODE? FORW</p> <p>Returns: 1 (the decimal form of "#01H")</p>
<b>Note(s)</b>	<HEXADECEMAL NUMERIC RESPONSE DATA>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication</p> <p>LINS10:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?</p>

---

## SCPI Command Reference

### FTFL/PT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier**

<b>Description</b>	<p>This command sets the Fault Type Fault Location (FTFL) Operator Identifier to be generated. At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; FTFL</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier &lt;wsp&gt;FORWard   BACKward, &lt;Identifier&gt;</p>
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWard   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p> <p><b>Identifier:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the operator identifier to be generated. A maximum of 9 characters are allowed.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU1:FTFL:IDEN FORW,"exfo"</p> <p>SOUR:DATA:TEL:OTN:ODU1:FTFL:IDEN? FORW</p> <p>Returns: "exfo"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication</p> <p>LINS10:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENTifier?**

<b>Description</b>	<p>This query returns the Fault Type Fault Location (FTFL) Operator Identifier to be generated. At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENTifier? <wsp>FORWARD   BACKward
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Identifier>
<b>Response(s)</b>	<p><b>Identifier:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the value of the Fault Type Fault Location (FTFL) Operator Identifier (bytes 1 to 9 for forward, byte 129 to 137 for backward) to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU1:FTFL:IDEN FORW,"exfo"</p> <p>SOUR:DATA:TEL:OTN:ODU1:FTFL:IDEN? FORW</p> <p>Returns: "exfo"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication</p> <p>LINS10:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?</p>

## SCPI Command Reference

### FTFL/PT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPec**

<b>Description</b>	<p>This command sets the Fault Type Fault Location (FTFL) Operator Specific to be generated. At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; FTFL</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPec &lt;wsp&gt;FORWard   BACKward, &lt;Specific&gt;</p>
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWard   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p> <p><b>Specific:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the operator specific to be generated. A maximum of 118 characters are allowed.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU1:FTFL:OPSP FORW,"exfo"</p> <p>SOUR:DATA:TEL:OTN:ODU1:FTFL:OPSP? FORW</p> <p>Returns: "exfo"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication</p> <p>LINS10:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?</p>

---



**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPec?**

<b>Description</b>	<p>This query returns the Fault Type Fault Location (FTFL) Operator Specific to be generated.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; FTFL</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPec? <wsp>FORWard   BACKward
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWard   BACKward</p> <p>Sets the configuration of the forward and backward ODU FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Specific>
<b>Response(s)</b>	<p><b>Specific:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the value of operator specific (bytes 10 to 127 for forward, byte 138 to 255 for backward) to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU1:FTFL:OPSP FORW,"exfo"</p> <p>SOUR:DATA:TEL:OTN:ODU1:FTFL:OPSP? FORW</p> <p>Returns: "exfo"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPec

## SCPI Command Reference

### FTFL/PT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication**

<b>Description</b>	<p>This command sets the FTFL Fault Indication message to be generated for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to NFAult.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; FTFL/PT &gt; ODU3e(1/2) &gt; FTFL</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; FTFL/PT &gt; ODU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication &lt;wsp&gt;FORWard   BACKward, NFAult   SFAil   SDEGrade   REServed</p>
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWard   BACKward</p> <p>Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p> <p><b>Indication:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NFAult   SFAil   SDEGrade   REServed</p> <p>Sets the FTFL Fault Indication message to be generated.</p> <p>NFAult: No Fault (NFAult)</p> <p>SFAil: Signal Fail (SFAil)</p> <p>SDEGrade: Signal Degraded (SDEGrade)</p> <p>REServed: Reserved</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:IND FORW,SFA</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:IND? FORW</p> <p>Returns: SFAIL</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication?</p>

<b>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication?</b>	
<b>Description</b>	<p>This query returns the FTFL Fault Indication message to be generated for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to NFAult.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; FTFL/PT &gt; ODU3e(1/2) &gt; FTFL</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; FTFL/PT &gt; ODU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication? <wsp>FORWARD   BACKward
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKward</p> <p>Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL). FORWARD, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Indication>
<b>Response(s)</b>	<p><b>Indication:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the FTFL Fault Indication message to be generated.</p> <p>NFAULT, No Fault (NFAULT) is selected as FTFL Fault Indication message.</p> <p>SFAIL, Signal Fail (SFAIL) is selected as FTFL Fault Indication message.</p> <p>SDEGRADE, Signal Degraded (SDEGRADE) is selected as FTFL Fault Indication message.</p> <p>RESERVED, Reserved is selected as FTFL Fault Indication message.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:IND FORW,SFA</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:IND? FORW</p> <p>Returns: SFAIL</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication

## SCPI Command Reference

### FTFL/PT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE**

<b>Description</b>	<p>This command sets the FTFL Fault Indication code to be generated for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to #H00.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; FTFL/PT &gt; ODU3e(1/2) &gt; FTFL</p> <p>At *RST condition, this value is set to NFAult.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; FTFL/PT &gt; ODU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE &lt;wsp&gt;FORWard   BACKward, &lt;Code&gt;</p>
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWard   BACKward</p> <p>Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p> <p><b>Code:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the FTFL Fault Indication Code to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:CODE FORW,#H01</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:CODE? FORW</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE?</p>

**:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE?**

<b>Description</b>	<p>This query returns the FTFL Fault Indication code to be generated for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to #H00.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; FTFL/PT &gt; ODU3e(1/2) &gt; FTFL</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; FTFL/PT &gt; ODU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE? <wsp>FORWARD   BACKWARD
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKWARD</p> <p>Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL). FORWARD, sets the Forward configuration.</p> <p>BACKWARD, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the selection of the Fault Type Fault Location (FTFL) fault indication code to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:CODE FORW,#H01</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:CODE? FORW</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:FTFL:CODE

## SCPI Command Reference

### FTFL/PT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENtifier**

<b>Description</b>	<p>This command sets the FTFL Operator Identifier to be generated for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; FTFL/PT &gt; ODU3e(1/2) &gt; FTFL</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; FTFL/PT &gt; ODU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENtifier &lt;wsp&gt;FORWARD   BACKward, &lt;Identifier&gt;</p>
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKward</p> <p>Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p> <p><b>Identifier:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the operator identifier to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:IDEN FORW,"exfo"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:IDEN? FORW</p> <p>Returns: "exfo"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENTifier?**

<b>Description</b>	<p>This query returns the FTFL Operator Identifier to be generated for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; FTFL/PT &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; FTFL/PT &gt; ODU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENTifier? &lt;wsp&gt;FORWARD   BACKward</p>
<b>Parameter(s)</b>	<p><b>Fttl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKward</p> <p>Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL). FORWARD, sets the Forward configuration. BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Identifier>
<b>Response(s)</b>	<p><b>Identifier:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the value of the Fault Type Fault Location (FTFL) Operator Identifier (bytes 1 to 9 for forward, byte 129 to 137 for backward) to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:IDEN FORW,"exfo"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:IDEN? FORW</p> <p>Returns: "exfo"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENTifier

---

## SCPI Command Reference

### FTFL/PT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:OPSPec**

<b>Description</b>	<p>This command sets the FTFL Operator Specific to be generated for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO ODU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; FTFL/PT &gt; ODU3e(1/2) &gt; FTFL.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; FTFL/PT &gt; ODU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:OPSPec &lt;wsp&gt;FORWard   BACKward, &lt;Specific&gt;</p>
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWard   BACKward</p> <p>Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p> <p><b>Specific:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the operator specific to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:OPSP FORW,"exfo"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:OPSP? FORW</p> <p>Returns: "exfo"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPec?</p>



---

**:SOURce[1..n]:DATA:TELeom:OTN:ODU[1..n]:E[1..n]:FTFL:OPSPec?**

<b>Description</b>	<p>This query returns the FTFL Operator Specific to be generated for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO ODU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU3e(1/2) &gt; FTFL/PT &gt; ODU3e(1/2) &gt; FTFL.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU4 &gt; FTFL/PT &gt; ODU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELeom:OTN:ODU[1..n]:E[1..n]:FTFL:OPSPec? <wsp>FORWard   BACKward
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWard   BACKward</p> <p>Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL). FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Specific>
<b>Response(s)</b>	<p><b>Specific:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the value of operator specific (bytes 10 to 127 for forward, byte 138 to 255 for backward) to be generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:OPSP FORW,"exfo"</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:FTFL:OPSP? FORW</p> <p>Returns: "exfo"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELeom:OTN:ODU[1..n]:FTFL:OPSPec

---

## SCPI Command Reference

### *FTFL/PT*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPE**

<b>Description</b>	<p>This command sets the injected payload signal type to be generated.</p> <p>At *RST condition, this value is set to PRBStest.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; Payload Type</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPE &lt;wsp&gt;RFSTANDARD   EXPERIMENTAL   ASYNCHRONOUS   BISYNCH   ATM   GFP1   VCONCATE   PCSCODEWORD   FC1200ODU2E   GFPEOPU2   OC3STM1ODU0   OC12STM4ODU0   FC100ODU0   FC200ODU1   FC400   FC800     BSTIMING   BSNTIMING   IBDRMAPPING   IBSDRMAPPING   IBQDRMAPPING   ODUODTUJK   ODUODTUKTSODTUJK   NAVAILABLE   RPROPRIET   NULLTEST   PRBSTEST</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe**

Parameter(s)	Payload Type:
	The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.
	The allowed elements for this parameter are: RFSTANDARD   EXPERIMENTAL   ASYNCHRONOUS   BISYNCH   ATM   GFP1   VCONCATE   PCSCODEWORD   FC1200ODU2E   GFPEOPU2   OC3STM1ODU0   OC12STM4ODU0   FC100ODU0   FC200ODU1   FC400   FC800     BSTIMING   BSNTIMING   IBDRMAPPING   IBSDRMAPPING   IBQDRMAPPING   ODUODTUJK   ODUODTUKTSODTUJK   NAVAILABLE   RPROPRIET   NULLTEST   PRBSTEST
	Sets the injected payload signal type to be generated.
	RFSTANDARD: Reserved for international standardization
	EXPERIMENTAL: Experimental mapping
	ASYNCHRONOUS: Asynchronous CBR mapping
	BISYNCH: Bit synchronous CBR mapping
	ATM: ATM mapping
	GFP1: GFP Mapping
	VCONCATE: Virtual Concatenated signal
	PCSCODEWORD: PCS Codeword transparent Ethernet
	FC1200ODU2E: FC-1200 into ODU2e
	GFPEOPU2: GFP mapping into extended OPU2
	OC3STM1ODU0: OC3/STM-1 into ODU0
	OC12STM4ODU0: OC12/STM-4 into ODU0
	FC100ODU0: FC-100 into ODU0
	FC200ODU1: FC-200 into ODU1
	FC400: FC-400 into ODUflex
	FC800: FC-800 into ODUflex
	BSTIMING: Bit stream with octet timing mapping
	BSNTIMING: Bit stream without octet timing mapping
	IBDRMAPPING: IB DDR mapping into ODUflex
	IBSDRMAPPING: IB SDR mapping into ODUflex
	IBQDRMAPPING: IB QDR mapping into ODUflex

## SCPI Command Reference

*FTFL/PT*

---

**:SOURce[1..n]:DATA:TELecom:OTN:OPU[1..n]:PTYPe**

<b>Parameter(s)</b>	ODUODTUIJK: ODU multiplex with ODTUjk ODUODTUKTSODTUIJK: ODU multiplex with ODTUk.ts/ODTUjk NAVAILABLE: Not available RPROPRIET: Reserved codes for proprietary use NULLTEST: NULL test signal mapping PRBSTEST: PRBS test signal mapping
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OPU1:PTYP EXP SOUR:DATA:TEL:OTN:OPU1:PTYP? Returns: EXPERIMENTAL
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:PTYPe SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:PTYPe?

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPE?**

---

<b>Description</b>	This query returns the injected payload signal type to be generated. At *RST condition, this value is set to PRBStest. Navigation Path: Test > Setup > OTN BERT > OTU > FTFL/PT > ODU > Payload Type
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPE?
<b>Response Syntax</b>	<Payload>

---

## SCPI Command Reference

FTFL/PT

:SOURCE[1..n]:DATA:TELECOM:OTN:OPU[1..n]:PTYPE?

Response(s)	Payload:
	The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.
	Returns the expected payload signal type to be generated.
	RFSTANDARD, indicates Reserved for international standardization
	EXPERIMENTAL, indicates Experimental mapping
	ASYNCHRONOUS, indicates Asynchronous CBR mapping
	BISYNCH, indicates Bit synchronous CBR mapping
	ATM, indicates ATM mapping
	GFP1, indicates GFP Mapping
	VCONCATE, indicates Virtual Concatenated signal
	PCSCODEWORD, indicates PCS Codeword transparent Ethernet
	FC1200ODU2E, indicates FC-1200 into ODU2e
	GFPEOPU2, indicates GFP mapping into extended OPU2
	OC3STM1ODU0, indicates OC3/STM-1 into ODU0
	OC12STM4ODU0, indicates OC12/STM-4 into ODU0
	FC100ODU0, indicates FC-100 into ODU0
	FC200ODU1, indicates FC-200 into ODU1
	FC400, indicates FC-400 into ODUflex
	FC800, indicates FC-800 into ODUflex
	BSTIMING, indicates Bit stream with octet timing mapping
	BSNTIMING, indicates Bit stream without octet timing mapping
	IBDRMAPPING, indicates IB DDR mapping into ODUflex
	IBSDRMAPPING, indicates IB SDR mapping into ODUflex
	IBQDRMAPPING, indicates IB QDR mapping into ODUflex
	ODUODTUJK, indicates ODU multiplex with ODTUjk
	ODUODTUKTSODTUJK, indicates ODU multiplex with ODTUk.ts/ODTUjk
	NAVAILABLE, indicates Not available
	RPROPRIET, indicates Reserved codes for proprietary use
	NULLTEST, indicates NULL test signal mapping
	PRBSTEST, indicates PRBS test signal mapping

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?**

<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OPU1:PTYP EXP SOUR:DATA:TEL:OTN:OPU1:PTYP? Returns: EXPERIMENTAL
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?

---

## SCPI Command Reference

### *FTFL/PT*

---

**:SOURce[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCODE**

<b>Description</b>	<p>This command sets the corresponding injected payload type as hexadecimal code. At *RST condition, this value is set to #H00. Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; Payload Type</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCODE &lt;wsp&gt; &lt;Code&gt;</p>
<b>Parameter(s)</b>	<p><b>Code:</b> The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element. Sets the corresponding injected payload type in hexadecimal code. The values are 00 to FF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OPU1:PCOD #H00 SOUR:DATA:TEL:OTN:OPU1:PCOD? Returns: 00 (the decimal form of "H00")</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCODE SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCODE?</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?**

<b>Description</b>	<p>This query returns the corresponding injected payload type as hexadecimal code.</p> <p>At *RST condition, this value is set to #H03.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; Payload Type</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the corresponding injected payload type as decimal code.</p> <p>The values are 00 to 255.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OPU1:PCOD #H00</p> <p>SOUR:DATA:TEL:OTN:OPU1:PCOD?</p> <p>Returns: 00 (the decimal form of "#H00")</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?</p>

---

## SCPI Command Reference

### FTFL/PT

---

:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe

#### Description

This command sets the expected Payload Type.

At \*RST condition, this value is set to PRBStest.

Navigation Path: Test > Setup > OTN BERT > OTU > FTFL/PT > ODU > Payload Type

Navigation Path: Test > Results > FTFL/PT > OPU > PT

In Multi-Channel OTN, when accessing an OPU LO, the channel must be set using SENS:DATA:TEL:OTN:OPU[1..n]CHAN. The expected Payload Type can also be set for all channels by using SENS:DATA:TEL:OTN:OPU[1..n]:ACHA.

#### Syntax

:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe <wsp>RFSTandard | EXPerimental | ASYNchronous | BISYNch | PCSCTE | ATM | GFP | VCONcate | PCSCTE | FC1200ODU2E | GFPEOPU2 | OC3STM1ODU0 | OC12STM4ODU0 | FC100ODU0 | FC200ODU1 | FC400ODUFLEX | FC800ODUFLEX | BSTiming | BSNTiming | IBSDRODUFlex | IBDDRODUFlex | IBQDRODUFlex | ODUODTUKTS | ODUODTUJK | NAVailable | RPRopriet | NULLtest | PRBStest | GFP1 | PCSCODEWORD | FC400 | FC800 | IBSDRMAPPING | IBDRMAPPING | IBQDRMAPPING | ODUODTUKTSODTUJK

---

:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPE

**Parameter(s)****Payload Type:**

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: RFSTandard | EXPerimental | ASYNchronous | BISYNch | PCSCTE | ATM | GFP | VCONcate | PCSCTE | FC1200ODU2E | GFPEOPU2 | OC3STM1ODU0 | OC12STM4ODU0 | FC100ODU0 | FC200ODU1 | FC400ODUFLEX | FC800ODUFLEX | BSTiming | BSNTiming | IBSDRODUFlex | IBDDRODUFlex | IBQDRODUFlex | ODUODTUKTS | ODUODTUJK | NAVailable | RPRopriet | NULLtest | PRBStest | GFP1 | PCSCODEWORD | FC400 | FC800 | IBSDRMAPPING | IBDRMAPPING | IBQDRMAPPING | ODUODTUKTSODTUJK

Sets the expected Payload Type.

RFSTANDARD: Reserved for international standardization

EXPERIMENTAL: Experimental mapping

ASYNCHRONOUS: Asynchronous CBR mapping

BISYNCH: Bit synchronous CBR mapping

ATM: ATM mapping

GFP1: GFP Mapping

VCONCATE: Virtual Concatenated signal

PCSCODEWORD: PCS Codeword transparent Ethernet

FC1200ODU2E: FC-1200 into ODU2e

GFPEOPU2: GFP mapping into extended OPU2

OC3STM1ODU0: OC3/STM-1 into ODU0

OC12STM4ODU0: OC12/STM-4 into ODU0

FC100ODU0: FC-100 into ODU0

FC200ODU1: FC-200 into ODU1

FC400: FC-400 into ODUFlex

FC800: FC-800 into ODUFlex

BSTIMING: Bit stream with octet timing mapping

BSNTIMING: Bit stream without octet timing mapping

IBDRMAPPING: IB DDR mapping into ODUFlex

IBSDRMAPPING: IB SDR mapping into ODUFlex

## SCPI Command Reference

*FTFL/PT*

---

**:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:PTYPe**

**Parameter(s)** IBQDRMAPPING: IB QDR mapping into ODUflex  
ODUODTUIJK: ODU multiplex with ODTUjk  
ODUODTUKTSODTUIJK: ODU multiplex with ODTUk.ts/ODTUjk  
NAVAILABLE: Not available  
RPROPRIET: Reserved codes for proprietary use  
NULLTEST: NULL test signal mapping  
PRBSTEST: PRBS test signal mapping

**Example(s)** OTN BERT:  
SENS:DATA:TEL:OTN:OPU1:PTYP EXP  
SENS:DATA:TEL:OTN:OPU1:PTYP?  
Returns: EXPERIMENTAL  
For Multi-Channel OTN:  
SENS:DATA:TEL:OTN:OPU100:ACHA ON  
SENS:DATA:TEL:OTN:OPU100:PTYP EXP  
SENS:DATA:TEL:OTN:OPU100:PTYP?  
Returns: EXPERIMENTAL

**See Also** SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCoDe  
SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCoDe?  
  
SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:CHAN  
SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:CHAN?  
  
SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:ACHA  
SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:ACHA?

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?**

<b>Description</b>	<p>This query returns the expected Payload Type.</p> <p>At *RST condition, this value is set to PRBStest.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; Payload Type</p> <p>Navigation Path: Test &gt; Results &gt; FTFL/PT &gt; OPU &gt; PT</p> <p>In Multi-Channel OTN, when accessing an OPU LO, the channel must be set using SENS:DATA:TEL:OTN:OPU[1..n]CHAN. The Payload Type message can also be set for all channels by using SENS:DATA:TEL:OTN:OPU[1..n]:ACHA.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?
<b>Response Syntax</b>	<Payload Type>

---

## SCPI Command Reference

FTFL/PT

:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPE?

### Response(s)

#### Payload Type:

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the expected Payload Type.

RFSTANDARD, indicates Reserved for international standardization

EXPERIMENTAL, indicates Experimental mapping

ASYNCHRONOUS, indicates Asynchronous CBR mapping

BISYNCH, indicates Bit synchronous CBR mapping

ATM, indicates ATM mapping

GFP1, indicates GFP Mapping

VCONCATE, indicates Virtual Concatenated signal

PCSCODEWORD, indicates PCS Codeword transparent Ethernet

FC1200ODU2E, indicates FC-1200 into ODU2e

GFPEOPU2, indicates GFP mapping into extended OPU2

OC3STM1ODU0, indicates OC3/STM-1 into ODU0

OC12STM4ODU0, indicates OC12/STM-4 into ODU0

FC100ODU0, indicates FC-100 into ODU0

FC200ODU1, indicates FC-200 into ODU1

FC400, indicates FC-400 into ODUflex

FC800, indicates FC-800 into ODUflex

BSTIMING, indicates Bit stream with octet timing mapping

BSNTIMING, indicates Bit stream without octet timing mapping

IBDRMAPPING, indicates IB DDR mapping into ODUflex

IBSDRMAPPING, indicates IB SDR mapping into ODUflex

IBQDRMAPPING, indicates IB QDR mapping into ODUflex

ODUODTUJK, indicates ODU multiplex with ODTUjk

ODUODTUKTSODTUJK, indicates ODU multiplex with ODTUk.ts/ODTUjk

NAVAILABLE, indicates Not available

RPROPRIET, indicates Reserved codes for proprietary use

NULLTEST, indicates NULL test signal mapping

PRBSTEST, indicates PRBS test signal mapping

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?**

---

**Example(s)**

OTN BERT:  
SENS:DATA:TEL:OTN:OPU1:PTYP EXP  
SENS:DATA:TEL:OTN:OPU1:PTYP?  
Returns: EXPERIMENTAL  
For Multi-Channel OTN:  
SENS:DATA:TEL:OTN:OPU100:ACHA ON  
SENS:DATA:TEL:OTN:OPU100:PTYP EXP  
SENS:DATA:TEL:OTN:OPU100:PTYP?  
Returns: EXPERIMENTAL

**See Also**

SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE  
SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?  
  
SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN  
SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN?  
  
SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA  
SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA?

---

## SCPI Command Reference

### FTFL/PT

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE**

<b>Description</b>	<p>This command sets the expected Payload Type as hexadecimal code.</p> <p>At *RST condition, this value is set to PRBStest.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; Payload Type</p> <p>Navigation Path: Test &gt; Results &gt; FTFL/PT &gt; OPU &gt; PT</p> <p>In Multi-Channel OTN, when accessing an OPU LO, the channel must be set using SENS:DATA:TEL:OTN:OPU[1..n]CHAN. The expected Payload Type can also be set for all channels by using SENS:DATA:TEL:OTN:OPU[1..n]:ACHA.</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE &lt;wsp&gt; &lt;Code&gt;</code>
<b>Parameter(s)</b>	<p><b>Code:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the corresponding expected payload type as hexadecimal code.</p>
<b>Example(s)</b>	<p>OTN BERT:</p> <pre>SENS:DATA:TEL:OTN:OPU1:PCOD #H00 SENS:DATA:TEL:OTN:OPU1:PCOD? Returns: 0 (the decimal form of "#H00")</pre> <p>For Multi-Channel OTN:</p> <pre>SENS:DATA:TEL:OTN:OPU100:ACHA ON SENS:DATA:TEL:OTN:OPU100:PCOD #H00 SENS:DATA:TEL:OTN:OPU100:PCOD? Returns: 0 (the decimal form of "#H00")</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?  SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN?  SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA?</pre>



---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?**

---

<b>Description</b>	<p>This query returns the expected Payload Type as hexadecimal code.</p> <p>At *RST condition, this value is set to PRBStest.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; Payload Type</p> <p>Navigation Path: Test &gt; Results &gt; FTFL/PT &gt; OPU &gt; PT</p> <p>In Multi-Channel OTN, when accessing an OPU LO, the channel must be set using SENS:DATA:TEL:OTN:OPU[1..n]CHAN. The expected Payload Type can also be set for all channels by using SENS:DATA:TEL:OTN:OPU[1..n]:ACHA.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?
<b>Response Syntax</b>	<Code>

---

## SCPI Command Reference

*FTFL/PT*

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?**

<b>Response(s)</b>	<b>Code:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the corresponding expected Payload Type as decimal code. The values are from 0 to 255.
<b>Example(s)</b>	OTN BERT: SENS:DATA:TEL:OTN:OPU1:PCOD #H00 SENS:DATA:TEL:OTN:OPU1:PCOD? Returns: 0 (the decimal form of "#H00") For Multi-Channel OTN: SENS:DATA:TEL:OTN:OPU100:ACHA ON SENS:DATA:TEL:OTN:OPU100:PCOD #H00 SENS:DATA:TEL:OTN:OPU100:PCOD? Returns: 0 (the decimal form of "#H00")
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?  SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN?  SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA?

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PLM**

<b>Description</b>	<p>This command enables or disables the state of Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis for the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; Payload Type</p> <p>Navigation Path: Test &gt; Results &gt; FTFL/PT &gt; OPU &gt; PT</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PLM <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the OPU-PLM (Optical Payload Unit-Payload Label Mismatch) alarm analysis.</p>
<b>Example(s)</b>	<p>OTN BERT:</p> <p>SENS:DATA:TEL:OTN:OPU1:PLM ON</p> <p>SENS:DATA:TEL:OTN:OPU1:PLM?</p> <p>Returns: 1</p> <p>For Multi-Channel OTN:</p> <p>SENS:DATA:TEL:OTN:OPU100:ACHA ON</p> <p>SENS:DATA:TEL:OTN:OPU100:PLM ON</p> <p>SENS:DATA:TEL:OTN:OPU100:PLM?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCoDe</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCoDe?</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN?</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA?</p>

## SCPI Command Reference

FTFL/PT

---

:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PLM?

<b>Description</b>	<p>This query returns the state of Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis for the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU &gt; FTFL/PT &gt; ODU &gt; Payload Type</p> <p>Navigation Path: Test &gt; Results &gt; FTFL/PT &gt; OPU &gt; PT</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PLM?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.</p> <p>1, PLM alarm analysis is enabled.</p> <p>0, PLM alarm analysis is disabled.</p>
<b>Example(s)</b>	<p>OTN BERT:</p> <p>SENS:DATA:TEL:OTN:OPU1:PLM ON</p> <p>SENS:DATA:TEL:OTN:OPU1:PLM?</p> <p>Returns: 1</p> <p>For Multi-Channel OTN:</p> <p>SENS:DATA:TEL:OTN:OPU100:ACHA ON</p> <p>SENS:DATA:TEL:OTN:OPU100:PLM ON</p> <p>SENS:DATA:TEL:OTN:OPU100:PLM?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCoDe</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCoDe?</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPE**

<b>Description</b>	<p>This command sets the injected payload signal type to be generated.</p> <p>At *RST condition, this value is set to PRBStest.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU 1/2 E&gt; FTFL/PT &gt; ODU &gt; Payload Type</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPE &lt;wsp&gt;RFSTANDARD   EXPERIMENTAL   ASYNCHRONOUS   BISYNCH   ATM   GFP1   VCONCATE   PCSCODEWORD   FC1200ODU2E   GFPEOPU2   OC3STM1ODU0   OC12STM4ODU0   FC100ODU0   FC200ODU1   FC400   FC800     BSTIMING   BSNTIMING   IBDRMAPPING   IBSDRMAPPING   IBQDRMAPPING   ODUODTUJK   ODUODTUKTSODTUJK   NAVAILABLE   RPROPRIET   NULLTEST   PRBSTEST</pre>

## SCPI Command Reference

FTFL/PT

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPE

Parameter(s)	Payload Type:
	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RFSTANDARD   EXPERIMENTAL   ASYNCHRONOUS   BISYNCH   ATM   GFP1   VCONCATE   PCSCODEWORD   FC1200ODU2E   GFPEOPU2   OC3STM1ODU0   OC12STM4ODU0   FC100ODU0   FC200ODU1   FC400   FC800     BSTIMING   BSNTIMING   IBDRMAPPING   IBSDRMAPPING   IBQDRMAPPING   ODUODTUJK   ODUODTUKTSODTUJK   NAVAILABLE   RPROPRIET   NULLTEST   PRBSTEST</p> <p>Sets the injected payload signal type to be generated.</p> <p>RFSTANDARD: Reserved for international standardization</p> <p>EXPERIMENTAL: Experimental mapping</p> <p>ASYNCHRONOUS: Asynchronous CBR mapping</p> <p>BISYNCH: Bit synchronous CBR mapping</p> <p>ATM: ATM mapping</p> <p>GFP1: GFP Mapping</p> <p>VCONCATE: Virtual Concatenated signal</p> <p>PCSCODEWORD: PCS Codeword transparent Ethernet</p> <p>FC1200ODU2E: FC-1200 into ODU2e</p> <p>GFPEOPU2: GFP mapping into extended OPU2</p> <p>OC3STM1ODU0: OC3/STM-1 into ODU0</p> <p>OC12STM4ODU0: OC12/STM-4 into ODU0</p> <p>FC100ODU0: FC-100 into ODU0</p> <p>FC200ODU1: FC-200 into ODU1</p> <p>FC400: FC-400 into ODUflex</p> <p>FC800: FC-800 into ODUflex</p> <p>BSTIMING: Bit stream with octet timing mapping</p> <p>BSNTIMING: Bit stream without octet timing mapping</p> <p>IBDRMAPPING: IB DDR mapping into ODUflex</p> <p>IBSDRMAPPING: IB SDR mapping into ODUflex</p> <p>IBQDRMAPPING: IB QDR mapping into ODUflex</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe**

---

<b>Parameter(s)</b>	ODUODTUJK: ODU multiplex with ODTUjk ODUODTUKTSODTUJK: ODU multiplex with ODTUk.ts/ODTUjk NAVAILABLE: Not available RPROPRIET: Reserved codes for proprietary use NULLTEST: NULL test signal mapping PRBSTEST: PRBS test signal mapping
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OPU1:E:PTYP EXP SOUR:DATA:TEL:OTN:OPU1:E:PTYP? Returns: EXPERIMENTAL
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?

---

## SCPI Command Reference

### *FTFL/PT*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPE?**

<b>Description</b>	This query returns the injected payload signal type to be generated. At *RST condition, this value is set to PRBStest. Navigation Path: Test > Setup > OTN BERT > OTU 1/2 E> FTFL/PT > ODU > Payload Type
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPE?
<b>Response Syntax</b>	<Payload>

---



:SOURCE[1..n]:DATA:TELECOM:OTN:OPU[1..n]:E:PTYPE?

**Response(s)****Payload:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the expected payload signal type to be generated.

RFSTANDARD, indicates Reserved for international standardization

EXPERIMENTAL, indicates Experimental mapping

ASYNCHRONOUS, indicates Asynchronous CBR mapping

BISYNCH, indicates Bit synchronous CBR mapping

ATM, indicates ATM mapping

GFP1, indicates GFP Mapping

VCONCATE, indicates Virtual Concatenated signal

PCSCODEWORD, indicates PCS Codeword transparent Ethernet

FC1200ODU2E, indicates FC-1200 into ODU2e

GFPEOPU2, indicates GFP mapping into extended OPU2

OC3STM1ODU0, indicates OC3/STM-1 into ODU0

OC12STM4ODU0, indicates OC12/STM-4 into ODU0

FC100ODU0, indicates FC-100 into ODU0

FC200ODU1, indicates FC-200 into ODU1

FC400, indicates FC-400 into ODUflex

FC800, indicates FC-800 into ODUflex

BSTIMING, indicates Bit stream with octet timing mapping

BSNTIMING, indicates Bit stream without octet timing mapping

IBDRMAPPING, indicates IB DDR mapping into ODUflex

IBSDRMAPPING, indicates IB SDR mapping into ODUflex

IBQDRMAPPING, indicates IB QDR mapping into ODUflex

ODUODTUJK, indicates ODU multiplex with ODTUjk

ODUODTUKTSODTUJK, indicates ODU multiplex with ODTUk.ts/ODTUjk

NAVAILABLE, indicates Not available

RPROPRIET, indicates Reserved codes for proprietary use

NULLTEST, indicates NULL test signal mapping

PRBSTEST, indicates PRBS test signal mapping

## SCPI Command Reference

*FTFL/PT*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe?**

<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OPU1:E:PTYP EXP SOUR:DATA:TEL:OTN:OPU1:E:PTYP? Returns: EXPERIMENTAL
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?

---

**:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe:RECeived?**

<b>Description</b>	<p>This query returns the received payload signal type to be generated.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; ODU 1/2 E &gt; Payload Type</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe:RECeived?
<b>Response Syntax</b>	<Payload>
<b>Response(s)</b>	<p><b>Payload:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the received payload type.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:OPU1:E:PTYP:REC?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE:RECeived?

---

## SCPI Command Reference

*FTFL/PT*

---

### **:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:GOVEverwrite?**

<b>Description</b>	This query returns if an expected Payload Type has been overwritten on at least one channel. Navigation Path: Test > Setup > ODU Channels > PT NOTE: This query is only available in Multi-Channel OTN.
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:GOVEverwrite?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<b>Set:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. 1: At least one expected Payload Type has been overwritten. 0: All expected Payload Types are the same as the Global expected Payload Type.
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU100:GOVE?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE  SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN?  SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA?

---

---

**:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:CHANnel**

---

<b>Description</b>	<p>This command sets the value of the channel used by the Payload Type.</p> <p>At *RST condition, this value is set to 1.</p> <p>When setting a channel value, the AChannel value will automatically be set to OFF.</p> <p>NOTE: This command is only available in Multi-Channel OTN.</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:CHANnel &lt;wsp&gt; &lt;Channel&gt;</code>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel used for Payload Type, when AllCHAnnel is not selected, in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:OTN:OPU100:ACHA? Returns: 1 SENS:DATA:TEL:OTN:OPU100:CHAN 2 SENS:DATA:TEL:OTN:OPU100:ACHA? Returns: 0</pre>
<b>See Also</b>	<pre>SENS:DATA:TEL:OTN:OPU100:CHAN? SENS:DATA:TEL:OTN:OPU100:ACHA</pre>

---

## SCPI Command Reference

*FTFL/PT*

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHANnel?**

<b>Description</b>	<p>This query returns the value of the channel used by the Payload Type.</p> <p>At *RST condition, this value is set to 1.</p> <p>When setting a channel value, the AChannel value will automatically be set to OFF.</p> <p>NOTE: This command is only available in Multi-Channel OTN.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHANnel?
<b>Response Syntax</b>	<Channel>
<b>Response(s)</b>	<p><b>Channel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the channel used for the Payload Type, when AllCHannel is 'OFF', in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU100:CHAN?
<b>See Also</b>	SENS:DATA:TEL:OTN:OPU100:CHAN SENS:DATA:TEL:OTN:OPU100:ACHA

---

---

**:SENSe[1..n]:DATA:TELeom:OTN:OPU[1..n]:ACHannel**

<b>Description</b>	<p>This command forces the Payload Type commands/queries to be applied on all channels (ON) or a single channel (OFF).</p> <p>At *RST condition, this value is set to OFF.</p> <p>NOTE: This command is only available in Multi-Channel OTN.</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELeom:OTN:OPU[1..n]:ACHannel &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>AllChannel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Selects if the Payload Type commands/queries are applied on all channels (ON) or a single channel (OFF).</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:OTN:OPU100:ACHA ON SENS:DATA:TEL:OTN:OPU100:ACHA OFF</pre>
<b>See Also</b>	<pre>SENS:DATA:TEL:OTN:OPU100:CHAN? SENS:DATA:TEL:OTN:OPU100:ACHA?</pre>

---

## SCPI Command Reference

*FTFL/PT*

---

**:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:ACHannel?**

<b>Description</b>	<p>This query returns if the Payload Type commands/queries are applied on all channels (ON) or a single channel (OFF).</p> <p>At *RST condition, this value is set to OFF.</p> <p>NOTE: This command is only available in Multi-Channel OTN.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:ACHannel?
<b>Response Syntax</b>	<AllChannel>
<b>Response(s)</b>	<p><b>AllChannel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns an indication that the Payload Type commands/queries are to be applied on all channels.</p> <p>1, OPU Payload Type commands/queries are applied on all channels.</p> <p>0, OPU Payload Type commands/queries are applied on a specific channels determined by SENS:DATA:TEL:OTN:OPU[1..n]:CHAN.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU100:ACHA?
<b>See Also</b>	SENS:DATA:TEL:OTN:OPU100:ACHA SENS:DATA:TEL:OTN:OPU100:CHAN

---



---

**:SENSe[1..n]:DATA:TELeom:OTN:OPU[1..n]:E:PTYPe**

---

**Description**

This command sets the expected payload signal type to be generated.

At \*RST condition, this value is set to PRBStest.

Navigation Path: Test > Setup > OTN BERT > OTU 1/2 E > FTFL/PT > Payload Type

**Syntax**

:SENSe[1..n]:DATA:TELeom:OTN:OPU[1..n]:E:PTYPe <wsp>RFSTandard |  
EXPerimental | ASYNchronous | BISYNch | PCSCTE | ATM | GFP | VCONcate | PCSCTE |  
FC1200ODU2E | GFPEOPU2 | OC3STM1ODU0 | OC12STM4ODU0 | FC100ODU0 |  
FC200ODU1 | FC400ODUFLEX | FC800ODUFLEX | BSTiming | BSNTiming |  
IBSDRODUFlex | IBDDRODUFlex | IBQDRODUFlex | ODUODTUKTS | ODUODTUJK |  
NAVailable | RPRopriet | NULLtest | PRBStest

---

## SCPI Command Reference

FTFL/PT

:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe

Parameter(s)	Payload:
	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RFSTandard   EXPerimental   ASYNchronous   BISYNch   PCSCTE   ATM   GFP   VCONcate   PCSCTE   FC1200ODU2E   GFPEOPU2   OC3STM1ODU0   OC12STM4ODU0   FC100ODU0   FC200ODU1   FC400ODUFLEX   FC800ODUFLEX   BSTiming   BSNTiming   IBSDRODUFlex   IBDDRODUFlex   IBQDRODUFlex   ODUODTUKTS   ODUODTUJK   NAVailable   RPRopriet   NULLtest   PRBStest</p> <p>Sets the expected payload signal type to be generated.</p> <p>RFSTANDARD: Reserved for international standardization</p> <p>EXPERIMENTAL: Experimental mapping</p> <p>ASYNCHRONOUS: Asynchronous CBR mapping</p> <p>BISYNCH: Bit synchronous CBR mapping</p> <p>ATM: ATM mapping</p> <p>GFP1: GFP Mapping</p> <p>VCONCATE: Virtual Concatenated signal</p> <p>PCSCODEWORD: PCS Codeword transparent Ethernet</p> <p>FC1200ODU2E: FC-1200 into ODU2e</p> <p>GFPEOPU2: GFP mapping into extended OPU2</p> <p>OC3STM1ODU0: OC3/STM-1 into ODU0</p> <p>OC12STM4ODU0: OC12/STM-4 into ODU0</p> <p>FC100ODU0: FC-100 into ODU0</p> <p>FC200ODU1: FC-200 into ODU1</p> <p>FC400: FC-400 into ODUflex</p> <p>FC800: FC-800 into ODUflex</p> <p>BSTIMING: Bit stream with octet timing mapping</p> <p>BSNTIMING: Bit stream without octet timing mapping</p> <p>IBDRMAPPING: IB DDR mapping into ODUflex</p> <p>IBSDRMAPPING: IB SDR mapping into ODUflex</p> <p>IBQDRMAPPING: IB QDR mapping into ODUflex</p>

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe**

---

<b>Parameter(s)</b>	ODUODTUJK: ODU multiplex with ODTUjk ODUODTUKTSODTUJK: ODU multiplex with ODTUk.ts/ODTUjk NAVAILABLE: Not available RPROPRIET: Reserved codes for proprietary use NULLTEST: NULL test signal mapping PRBSTEST: PRBS test signal mapping
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU1:E:PTYP EXP SENS:DATA:TEL:OTN:OPU1:E:PTYP? Returns: EXPERIMENTAL
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?

---

## SCPI Command Reference

### *FTFL/PT*

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPE?**

<b>Description</b>	This query returns the expected payload signal type to be generated. At *RST condition, this value is set to PRBStest. Navigation Path: Test > Setup > OTN BERT > OTU 1/2 E > FTFL/PT > ODU > Payload Type
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPE?
<b>Response Syntax</b>	<Payload>

---

:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:E:PTYPe?

**Response(s)****Payload:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the expected payload signal type to be generated.

RFSTANDARD, indicates Reserved for international standardization

EXPERIMENTAL, indicates Experimental mapping

ASYNCHRONOUS, indicates Asynchronous CBR mapping

BISYNCH, indicates Bit synchronous CBR mapping

ATM, indicates ATM mapping

GFP1, indicates GFP Mapping

VCONCATE, indicates Virtual Concatenated signal

PCSCODEWORD, indicates PCS Codeword transparent Ethernet

FC1200ODU2E, indicates FC-1200 into ODU2e

GFPEOPU2, indicates GFP mapping into extended OPU2

OC3STM1ODU0, indicates OC3/STM-1 into ODU0

OC12STM4ODU0, indicates OC12/STM-4 into ODU0

FC100ODU0, indicates FC-100 into ODU0

FC200ODU1, indicates FC-200 into ODU1

FC400, indicates FC-400 into ODUflex

FC800, indicates FC-800 into ODUflex

BSTIMING, indicates Bit stream with octet timing mapping

BSNTIMING, indicates Bit stream without octet timing mapping

IBDRMAPPING, indicates IB DDR mapping into ODUflex

IBSDRMAPPING, indicates IB SDR mapping into ODUflex

IBQDRMAPPING, indicates IB QDR mapping into ODUflex

ODUODTUJK, indicates ODU multiplex with ODTUjk

ODUODTUKTSODTUJK, indicates ODU multiplex with ODTUk.ts/ODTUjk

NAVAILABLE, indicates Not available

RPROPRIET, indicates Reserved codes for proprietary use

NULLTEST, indicates NULL test signal mapping

PRBSTEST, indicates PRBS test signal mapping

## SCPI Command Reference

*FTFL/PT*

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPe?**

<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU1:E:PTYP EXP SENS:DATA:TEL:OTN:OPU1:E:PTYP? Returns: EXPERIMENTAL
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?

---

**:SOURce[1..n]:DATA:TELecom:OTN:OPU[1..n]:E:PCODE**

<b>Description</b>	<p>This command sets the corresponding injected payload type as hexadecimal code.</p> <p>At *RST condition, this value is set to #H00.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU 1/2 E&gt; FTFL/PT &gt; Payload Type</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELecom:OTN:OPU[1..n]:E:PCODE &lt;wsp&gt; &lt;Code&gt;</code>
<b>Parameter(s)</b>	<p><b>Code:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the corresponding injected payload type in hexadecimal code.</p> <p>The values are 00 to FF.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:OPU1:E:PCOD #H00 SOUR:DATA:TEL:OTN:OPU1:E:PCOD? Returns: 00 (the decimal form of "H00")</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCODE SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCODE?</pre>

---

## SCPI Command Reference

### FTFL/PT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE?**

<b>Description</b>	<p>This query returns the corresponding injected payload type as hexadecimal code.</p> <p>At *RST condition, this value is set to #H03.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU 1/2 E&gt; FTFL/PT &gt; Payload Type</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE?
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the corresponding injected payload type as decimal code.</p> <p>The values are 00 to 255.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OPU1:E:PCOD #H00</p> <p>SOUR:DATA:TEL:OTN:OPU1:E:PCOD?</p> <p>Returns: 00 (the decimal form of "H00")</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE</p> <p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?</p>

---



---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE**

<b>Description</b>	<p>This command sets the corresponding expected payload type as hexadecimal code.</p> <p>At *RST condition, this value is set to 03.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU 1/2 E&gt; FTFL/PT &gt; ODU &gt; Payload Type</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE &lt;wsp&gt; &lt;Code&gt;</code>
<b>Parameter(s)</b>	<p><b>Code:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the corresponding expected payload type as hexadecimal code.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OPU1:E:PCOD #H00</p> <p>SENS:DATA:TEL:OTN:OPU1:E:PCOD?</p> <p>Returns: 00 (the decimal form of "#H00")</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?</p>

---

## SCPI Command Reference

### FTFL/PT

---

---

#### :SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE?

---

<b>Description</b>	<p>This query returns the corresponding expected payload type as hexadecimal code.</p> <p>At *RST condition, this value is set to #HFE.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU 1/2 E &gt; FTFL/PT &gt; ODU &gt; Payload Type</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE?
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the corresponding expected payload type as decimal code.</p> <p>The values are from 0 to 255.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OPU1:E:PCOD #H00</p> <p>SENS:DATA:TEL:OTN:OPU1:E:PCOD?</p> <p>Returns: 00 (the decimal form of "H00")</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PLM**

<b>Description</b>	<p>This command enables or disables the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU 1/2 E &gt; FTFL/PT &gt; ODU &gt; Payload Type</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PLM <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the OPU-PLM (Optical Payload Unit-Payload Label Mismatch) alarm analysis.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OPU1:E:PLM ON</p> <p>SENS:DATA:TEL:OTN:OPU1:E:PLM?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?</p>

---

## SCPI Command Reference

*FTFL/PT*

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PLM?**

<b>Description</b>	<p>This query returns the status of the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU 1/2 E&gt; FTFL/PT &gt; ODU &gt; Payload Type</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PLM?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.</p> <p>1, PLM alarm analysis is enabled.</p> <p>0, PLM alarm analysis is disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OPU1:E:PLM ON</p> <p>SENS:DATA:TEL:OTN:OPU1:E:PLM?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?</p>

---

**:FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:E:PCODE:RECeived?**

---

<b>Description</b>	<p>This query returns the corresponding received payload type as decimal code.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; ODU 1/2E &gt; Payload Type</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:E:PCODE:RECeived?
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the received payload code.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:OPU1:E:PCOD:REC?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:PTYPe:RECeived?

---

## SCPI Command Reference

*FTFL/PT*

---

**:SENSe[1..n]:DATA:TELeom:OTN:OPU[1..n]:E:COPI**

<b>Description</b>	<p>This command sets the copy Rx to instrument.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Setup &gt; OTN BERT &gt; OTU 1/2 E&gt; FTFL/PT &gt; ODU &gt; Payload Type&gt;Copy RX</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:OTN:OPU[1..n]:E:COPI
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU1:E:COPI
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:COPI

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE**

---

<b>Description</b>	<p>This command selects the type of Optical Payload Unit (OPU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU 1/2 E) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE &lt;wsp&gt;MSIM   OAIS   OCSF</code>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MSIM   OAIS   OCSF</p> <p>Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OMSIM: OPU-OMSIM</p> <p>OAIS: OPU-OAIS</p> <p>OCSF: OPU-OCSF</p>
<b>Example(s)</b>	<p><code>SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OAIS</code></p> <p><code>SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE?</code></p> <p>Returns: OAIS</p>
<b>See Also</b>	<p><code>SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE</code></p> <p><code>SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?</code></p>

---

# MAC/IP/UDP

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination**

<b>Description</b>	<p>This command sets the Media Access Control (MAC) destination address for the selected traffic stream.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Ethernet &gt; Destination MAC Address</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination &lt;wsp&gt; &lt;Tgen&gt;, &lt;Address&gt;</pre>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Set the destination MAC address.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:ADDR:DEST 1, "FE:FE:FE:00:00:00" SOUR:DATA:TEL:ETH:STR:ADDR:DEST? 1 Returns: "FE:FE:FE:00:00:00"</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination? SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce</pre>



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination?**

<b>Description</b>	<p>This query returns the MAC destination address for the selected traffic stream.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Ethernet &gt; Destination MAC Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the MAC destination address in the form of a string.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:ADDR:DEST 1, "FE:FE:FE:00:00:00"</p> <p>SOUR:DATA:TEL:ETH:STR:ADDR:DEST? 1</p> <p>Returns: "FE:FE:FE:00:00:00"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce</p>

---

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination:IP**

<b>Description</b>	<p>This command sets the IP destination address for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.0.0.0.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPv4 &gt; Destination IP Address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination:IP &lt;wsp&gt; &lt;Tgen&gt;, &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Set the destination IP address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:ADDR:DEST:IP 1, "1.1.1.1"</p> <p>SOUR:DATA:TEL:ETH:STR:ADDR:DEST:IP? 1</p> <p>Returns: "1.1.1.1"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination:IP?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP?**


---

<b>Description</b>	<p>This query returns the IP destination address for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.0.0.0.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPv4 &gt; Destination IP Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the IP destination address in the form of a string.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:ADDR:DEST:IP 1, "1.1.1.1"</p> <p>SOUR:DATA:TEL:ETH:STR:ADDR:DEST:IP? 1</p> <p>Returns: "1.1.1.1"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP</p>

---

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce?

---

<b>Description</b>	<p>This query returns the Media Access Control (MAC) source address for the selected traffic stream.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Ethernet &gt; Source MAC Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the source MAC address in the form of a string.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:ADDR:SOUR 1, "FE:FE:FE:FE:00:00"</p> <p>SOUR:DATA:TEL:ETH:STR:ADDR:SOUR? 1</p> <p>Returns: "FE:FE:FE:FE:00:00"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:IP**

<b>Description</b>	<p>This command sets the IP source address for the selected traffic stream.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPv4 &gt; Source IP Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:IP <wsp> <Tgen>, <Address>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the source IP address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:ADDR:SOUR:IP 1, "1.1.1.1"</p> <p>SOUR:DATA:TEL:ETH:STR:ADDR:SOUR:IP? 1</p> <p>Returns: "1.1.1.1"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:IP?

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:IP?

---

<b>Description</b>	<p>This query returns the source IP address for the selected traffic stream.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPv4 &gt; Source IP Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:IP? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the source IP address in the form of a string.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:ADDR:SOUR:IP 1, "1.1.1.1"</p> <p>SOUR:DATA:TEL:ETH:STR:ADDR:SOUR:IP? 1</p> <p>Returns: "1.1.1.1"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:IP

---

<b>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve</b>	
<b>Description</b>	<p>This command sends a request to the network to retrieve the MAC address corresponding to the selected destination IP address.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Ethernet &gt; Destination MAC Address &gt; Resolve MAC Address</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPv4 &gt; Resolve MAC Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve <wsp> <Tgen>, ON   OFF
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the Resolved MAC address status.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DEST:RES 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:DEST:RES? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve?

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve?**

<b>Description</b>	<p>This query returns the status of the resolved MAC destination address (ARP).</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Ethernet &gt; Destination MAC Address &gt; Resolve MAC Address</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPv4 &gt; Resolve MAC Address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve? &lt;wsp&gt; &lt;Tgen&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Set&gt;</p>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of resolved MAC destination address.</p> <p>1, resolved MAC destination address is Enabled.</p> <p>0, resolved MAC destination address is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DEST:RES 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:DEST:RES? 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve</p>



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve:STATus?**

<b>Description</b>	<p>This query returns the status of the resolved MAC address. When enabled, will send an ARP request to the network to retrieve the MAC address corresponding to the selected IP address.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP IPv4 &gt; Resolve MAC Address</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve:STATus? &lt;wsp&gt; &lt;Tgen&gt;</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<code>&lt;Status&gt;</code>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the resolved MAC address status.</p> <p>FAILED, Failed is retrieved.</p> <p>NRESOLVED, Not Resolved is retrieved.</p> <p>RESOLVED, Resolved is retrieved.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:STR:DEST:RES 1, ON SOUR:DATA:TEL:ETH:STR:DEST:RES:STAT? 1</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:RESolve</code>

---

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:QPING

---

<b>Description</b>	This command quick pings with parameters for mentioned streams. Navigation Path: Test Setup > Test Configurator > MAC/IP/UDP > IP> Quick Ping
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:QPING <wsp> <Tgen>
<b>Parameter(s)</b>	<b>Tgen:</b> The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. EtherSAM: the service from 1 to 10. Traffic Gen and Mon: the stream from 1 to 16
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:QPING 1
<b>See Also</b>	SOUR[1..n]:DATA:TEL:ETH:STR:DEST:RES 1, ON

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:QPING?**

<b>Description</b>	This query returns the status of last Quick ping done.
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:QPING?
<b>Response Syntax</b>	<Ping Status>
<b>Response(s)</b>	<b>Ping Status:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Quick ping status. INPROGRESS   SUCCESSFUL   FAIL   NOPINGDONE
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:QPING 1 SOUR:DATA:TEL:ETH:STR:QPING?
<b>See Also</b>	SOUR[1..n]:DATA:TEL:ETH:STR:DEST:RES:STAT? 1

---

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ETHer?

---

<b>Description</b>	<p>This query returns the ether type for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0x0800.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Ethernet &gt; Ether Type</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ETHer? <wsp> <Set>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the ether type in form of short.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:ETH? 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IP:AUTomatic:STATus**

<b>Description</b>	<p>This command enables or disables the IP Automatic (DHCP) status for the selected traffic stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPV4 &gt; Automatic IP (DHCP)</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IP:AUTomatic:STATus &lt;wsp&gt; &lt;Tgen&gt;, ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the IP Automatic status for the selected stream.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:IP:AUT:STAT 1, ON SOUR:DATA:TEL:ETH:STR:IP:AUT:STAT? 1 Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL</pre>

---

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IP:AUTomatic:STATus?

---

<b>Description</b>	<p>This query returns the IP Automatic (DHCP) status for the selected traffic stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPV4 &gt; Automatic IP (DHCP)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IP:AUTomatic:STATus? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Automatic IP.</p> <p>1, Automatic IP is enabled.</p> <p>0, Automatic IP is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:IP:AUT:STAT 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:IP:AUT:STAT? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IP:AUTomatic:STATus

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway**

<b>Description</b>	<p>This command enables or disables the IP gateway status for the selected traffic stream.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPV4 &gt; Default Gateway</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway <wsp> <Tgen>, ON   OFF
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the IP gateway for the selected stream.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DEST:IP:GAT 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:DEST:IP:GAT? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway?

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway?**

<b>Description</b>	<p>This query returns the IP gateway status for the selected traffic stream.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPV4 &gt; Default Gateway</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway? &lt;wsp&gt; &lt;Tgen&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Set&gt;</p>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the IP gateway.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DEST:IP:GAT 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:DEST:IP:GAT? 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway</p>



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDRess**

<b>Description</b>	<p>This command sets the default gateway IP address for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.0.0.0.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPV4 &gt; Default Gateway</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDRess &lt;wsp&gt; &lt;Tgen&gt;, &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the default IP gateway address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DEST:IP:GAT 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:DEST:IP:GAT:ADDR 1, "0.0.1.1"</p> <p>SOUR:DATA:TEL:ETH:STR:DEST:IP:GAT:ADDR? 1</p> <p>Returns: "0.0.1.1"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDRess?</p>

---

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDRess?**

<b>Description</b>	<p>This query returns the default gateway IP address for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.0.0.0.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPV4 &gt; Default Gateway</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDRess? &lt;wsp&gt; &lt;Tgen&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Address&gt;</p>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the default gateway IP address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DEST:IP:GAT 1, ON SOUR:DATA:TEL:ETH:STR:DEST:IP:GAT:ADDR 1, "0.0.1.1" SOUR:DATA:TEL:ETH:STR:DEST:IP:GAT:ADDR? 1 Returns: "0.0.1.1"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDRess</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL**

<b>Description</b>	<p>This command sets the Time to Live (TTL) value for the selected traffic stream.</p> <p>At *RST condition, this value is set to 128.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPV4 &gt; TTL</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL <wsp> <Tgen>, <Ttl>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Ttl:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the TTL value.</p> <p>Choices are 0 to 255.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DEST:IP:TTL 1, 200</p> <p>SOUR:DATA:TEL:ETH:STR:DEST:IP:TTL? 1</p> <p>Returns: 200</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL?

## SCPI Command Reference

### MAC/IP/UDP

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL?

<b>Description</b>	<p>This query returns the Time to Live (TTL) value for the selected traffic stream.</p> <p>At *RST condition, this value is set to 128.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPV4 &gt; TTL</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL? <wsp> <Tgen>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the TTL value.</p> <p>This parameter is optional. If no token is specified, the current TTL value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p> <p>MINimum is used to retrieve the instrument's smallest supported TTL value.</p>
<b>Response Syntax</b>	<Ttl>
<b>Response(s)</b>	<p><b>Ttl:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the TTL value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DEST:IP:TTL 1, 200</p> <p>SOUR:DATA:TEL:ETH:STR:DEST:IP:TTL? 1</p> <p>Returns: 200</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TTL

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:MASK:IP**

<b>Description</b>	<p>This command sets the IP subnet mask for the selected traffic stream.</p> <p>At *RST condition, this value is set to 255.255.0.0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPV4 &gt; Subnet Mask</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:MASK:IP &lt;wsp&gt; &lt;Tgen&gt;, &lt;Mask&gt;</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Mask:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the IP subnet mask in the form of a string.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:SOUR:MASK:IP 1, "255.255.255.255" SOUR:DATA:TEL:ETH:STR:SOUR:MASK:IP? 1 Returns: "255.255.255.255"</pre>
<b>See Also</b>	<pre>SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:SOURCE:MASK:IP? SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDRESS</pre>

---

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:MASK:IP?**

<b>Description</b>	<p>This query returns the IP subnet mask for the selected traffic stream.</p> <p>At *RST condition, this value is set to 255.255.0.0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IPV4 &gt; Subnet Mask</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:MASK:IP? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the IP mask address in form of a string.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:SOUR:MASK:IP 1, "255.255.255.255"</p> <p>SOUR:DATA:TEL:ETH:STR:SOUR:MASK:IP? 1</p> <p>Returns: "255.255.255.255"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:MASK:IP</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:GATeway:ADDRESS</p>

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID**

---

<b>Description</b>	<p>This command sets the Virtual Local Area Network (VLAN) Identification (ID) of the stream.</p> <p>At *RST condition, this value is set to 2.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Vlan &gt; VLAN ID</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID &lt;wsp&gt; &lt;Tgen&gt;, &lt;Stacked&gt;, &lt;Vlanid&gt;   MAXimum   MINimum</p>

---

## SCPI Command Reference

MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID**

<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Stacked:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the Virtual Local Area Network (VLAN) stacked.</p> <p>The value for stacked is set to 1 only.</p> <p><b>Vlanid:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Virtual Local Area Network (VLAN) ID of the stream.</p> <p>Choices are 0 to 4095.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:VLAN 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:ID 1, 1, 50</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:ID? 1, 1</p> <p>Returns: 50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID?</p>

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID?**

<b>Description</b>	<p>This query returns the Virtual Local Area Network (VLAN) Identification (ID) of the stream. At *RST condition, this value is set to 2.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Vlan &gt; VLAN ID</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID? <wsp> <Tgen>, <Stacked>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Stacked:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Virtual Local Area Network (VLAN) stacked.</p> <p>The value for stacked is set to 1 only.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current VLAN ID is returned.</p> <p>Choices are 0 to 4095.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Vlanid>

## SCPI Command Reference

MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID?**

<b>Response(s)</b>	<b>Vlanid:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Virtual Local Area Network (VLAN) ID of the stream.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:VLAN 1, ON SOUR:DATA:TEL:ETH:STR:VLAN:ID 1, 1, 50 SOUR:DATA:TEL:ETH:STR:VLAN:ID? 1, 1 Returns: 50
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE**

---

**Description**

This command selects the type of Virtual Local Area Network (VLAN).

At \*RST condition, this value is set to device-dependent.

Navigation Path: BERT > Test Setup > Test Configurator > MAC/IP/UDP > Vlan > Type

**Syntax**

:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE <wsp> <Tgen>,  
<Stacked>, V8100 | V88A8 | V9100 | V9200 | V9300

---

## SCPI Command Reference

MAC/IP/UDP

---

:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE

<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Stacked:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p> <p><b>Vtype:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: V8100   V88A8   V9100   V9200   V9300</p> <p>Selects the VLAN type.</p> <p>V8100: 8100</p> <p>V88A8: 88A8</p> <p>V9100: 9100</p> <p>V9200: 9200</p> <p>V9300: 9300</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:VLAN 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:STAC 1, 3</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:TYPE 1, 1, V9100</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:TYPE? 1, 3</p> <p>Returns: V9100</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACed</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE?**

<b>Description</b>	<p>This query returns the type of the Virtual Local Area Network (VLAN).</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Vlan &gt; Type</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE? &lt;wsp&gt; &lt;Tgen&gt;, &lt;Stacked&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Stacked:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p>
<b>Response Syntax</b>	<p>&lt;Vtype&gt;</p>

---

## SCPI Command Reference

MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE?**

<b>Response(s)</b>	<b>Vtype:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of the VLAN. V8100, the VLAN type 8100 is selected. V88A8, the VLAN type 88A8 is selected. V9100, the VLAN type 9100 is selected. V9200, the VLAN type 9200 is selected. V9300, the VLAN type 9300 is selected.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:VLAN 1, ON SOUR:DATA:TEL:ETH:STR:VLAN:STAC 1, 3 SOUR:DATA:TEL:ETH:STR:VLAN:TYPE 1, 1, V9100 SOUR:DATA:TEL:ETH:STR:VLAN:TYPE? 1, 1 Returns: V9100
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACKed SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:PRiority**

<b>Description</b>	<p>This command sets the Virtual Local Area Network (VLAN) user priority.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Vlan &gt; Priority</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:PRiority <wsp> <Tgen>, <Stacked>, <Priority>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Stacked:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the Virtual Local Area Network (VLAN) stacked.</p> <p>The value for stacked is set to 1 only.</p> <p><b>Priority:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Virtual Local Area Network (VLAN) user priority.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:VLAN 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:PRI 1, 1, 5</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:PRI? 1, 1</p> <p>Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:PRiority?</p>

## SCPI Command Reference

MAC/IP/UDP

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:PRIority?

<b>Description</b>	<p>This query returns the Virtual Local Area Network (VLAN) user priority.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Vlan &gt; Priority</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:PRIority? &lt;wsp&gt; &lt;Tgen&gt;, &lt;Stacked&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Stacked:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the Virtual Local Area Network (VLAN) stacked.</p> <p>The value for stacked is set to 1 only.</p>
<b>Response Syntax</b>	<p>&lt;Priority&gt;</p>
<b>Response(s)</b>	<p><b>Priority:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Virtual Local Area Network (VLAN) user priority.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:VLAN 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:PRI 1, 1, 5</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:PRI? 1, 1</p> <p>Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:PRIority</p>



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID:ELIGiblebit**

<b>Description</b>	<p>This command enables or disables the Virtual Local Area Network (VLAN) Identification (ID) eligible bit.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Vlan &gt; Drop Eligible</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID:ELIGiblebit <wsp> <Tgen>, <Stacked>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Stacked:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Virtual Local Area Network (VLAN) stacked.</p> <p><b>Set:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables the eligible bit for the specific VLAN (Virtual Local Area Network) ID (Identification).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:VLAN 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:ID:ELIG 1, 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:ID:ELIG? 1, 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID:ELIGiblebit?</p>

## SCPI Command Reference

### MAC/IP/UDP

---

<b>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID:ELIGiblebit?</b>	
<b>Description</b>	<p>This query returns the status of Virtual Local Area Network (VLAN) Identification (ID) eligible bit.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Vlan &gt; Drop Eligible</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID:ELIGiblebit? &lt;wsp&gt; &lt;Tgen&gt;, &lt;Stacked&gt;</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Stacked:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Virtual Local Area Network (VLAN) stacked.</p>
<b>Response Syntax</b>	<code>&lt;Set&gt;</code>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Virtual Local Area Network (VLAN) ID eligible bit.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:VLAN 1, ON SOUR:DATA:TEL:ETH:STR:VLAN:ID:ELIG 1, 1, ON SOUR:DATA:TEL:ETH:STR:VLAN:ID:ELIG? 1, 1 Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:TYPE SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:ID:ELIGiblebit</pre>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT**

<b>Description</b>	<p>This command sets the UDP destination port number for the selected traffic stream.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; UDP &gt; Destination Port</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT &lt;wsp&gt; &lt;Tgen&gt;, &lt;Dport&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Dport:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the UDP destination port value for selected stream.</p> <p>Choices are 0 to 65535.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:DEST:PORT 1, 60 SOUR:DATA:TEL:ETH:STR:DEST:PORT? 1 Returns: 60</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT? SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT?</pre>

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT?

---

<b>Description</b>	<p>This query returns the UDP destination port number for the selected traffic stream.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; UDP &gt; Destination Port</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT? &lt;wsp&gt; &lt;Tgen&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the UDP destination port value for the selected stream.</p> <p>This parameter is optional. If no token is specified, the current UDP destination port number is returned.</p> <p>Choices are 0 to 65535.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Dport&gt;</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT?**

<b>Response(s)</b>	<b>Dport:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns UDP destination port number for the selected stream.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:DEST:PORT 1, 60 SOUR:DATA:TEL:ETH:STR:DEST:PORT? 1 Returns: 60
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT?

---

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT**

<b>Description</b>	<p>This command sets the UDP source port number for the selected traffic stream.</p> <p>At *RST condition, this value is set to 49184.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; UDP &gt; Source Port</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT &lt;wsp&gt; &lt;Tgen&gt;, &lt;Tport&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Tport:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the UDP source port number.</p> <p>Choices are 0 to 65535.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:SOUR:PORT 1, 65500</p> <p>SOUR:DATA:TEL:ETH:STR:SOUR:PORT? 1</p> <p>Returns: 65500</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT?**

<b>Description</b>	<p>This query returns the UDP source port number for the selected traffic stream.</p> <p>At *RST condition, this value is set to 49184.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; UDP &gt; Source Port</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT? &lt;wsp&gt; &lt;Tgen&gt;[, MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the UDP source port number.</p> <p>This parameter is optional. If no token is specified, the current source port number is returned.</p> <p>Choices are 0 to 65535.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Address&gt;</code>

---

## SCPI Command Reference

MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT?**

<b>Response(s)</b>	<b>Address:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the UDP source port number for the selected stream.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:SOUR:PORT 1, 65500 SOUR:DATA:TEL:ETH:STR:SOUR:PORT? 1 Returns: 65500
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:PORT SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:PORT:TCP?

---



**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs**

<b>Description</b>	<p>This command sets the Type of Service/Differentiated Services (TOS/DS) value for the selected traffic stream.</p> <p>At *RST condition, this value is set to 00.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs &lt;wsp&gt; &lt;Tgen&gt;, &lt;Tosds&gt;</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Tosds:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the TOS/DS value.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:DEST:IP:TOSD 1, #HDD SOUR:DATA:TEL:ETH:STR:DEST:IP:TOSD? 1 Returns: 221</pre>
<b>See Also</b>	<code>SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs?</code>

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs?

---

<b>Description</b>	<p>This query returns the Type of Service/Differentiated Services (TOS/DS) value for the selected traffic stream.</p> <p>At *RST condition, this value is set to 00.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Tosds>
<b>Response(s)</b>	<p><b>Tosds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the TOSDS value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DEST:IP:TOSD 1, #HDD</p> <p>SOUR:DATA:TEL:ETH:STR:DEST:IP:TOSD? 1</p> <p>Returns: 221</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IP:TOSDs

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:MULTiplicat**

<b>Description</b>	<p>This command enables or disables the IPv4 multiplier for the selected traffic stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt;IPv4 &gt; Source IP Multiplier</p> <p>Navigation Path: Test &gt; EtherSAM &gt; Test Configurator&gt; Set Up &gt;Services&gt;MAC/IP/UDP &gt;IPv4 &gt; Source IP Multiplier</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:MULTiplicat &lt;wsp&gt; &lt;Stream&gt;, ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the IP multiplier for the selected traffic stream.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:SOUR:IP:MULT 1, ON SOUR:DATA:TEL:ETH:STR:SOUR:IP:MULT? 1 Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IPVersion:HOP:LIMit?</pre>

---

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:MULTiplicat?

---

<b>Description</b>	<p>This query returns the status of IPv4 multiplier for the selected traffic stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt;IPv4 &gt; Source IP Multiplier</p> <p>Navigation Path: Test &gt; EtherSAM &gt; Test Configurator&gt; Set Up &gt;Services&gt;MAC/IP/UDP &gt;IPv4 &gt; Source IP Multiplier</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:MULTiplicat? <wsp><Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<IPv4 Multiplier Status>
<b>Response(s)</b>	<p><b>IPv4 Multiplier Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of IPv4 multiplier.</p> <p>1, IPv4 multiplier is Enabled.</p> <p>0, IPv4 multiplier is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:SOUR:IP:MULT 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:SOUR:IP:MULT? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IPVersion:HOP:LIMit

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:RANGe**

<b>Description</b>	<p>This command sets the IPv4 multiplier range for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0To127.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon &gt; Test Configurator&gt; Set Up &gt;Streams &gt;MAC/IP/UDP&gt;IPv4 &gt; Source IP Multiplier</p> <p>Navigation Path: Test EtherSAM &gt; Test Configurator&gt; Set Up &gt;Services &gt;MAC/IP/UDP&gt;IPv4 &gt; Source IP Multiplier</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:RANGe <wsp> <Stream>, 0T127   1T128
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Range:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 0T127   1T128</p> <p>Sets the range for IP multiplier.</p> <p>0T127: the range from 0 to 127.</p> <p>1T128: the range from 1 to 128.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:SOUR:IP:MULT 1, ON SOUR:DATA:TEL:ETH:STR:SOUR:IP:RANG 1, 1T128 SOUR:DATA:TEL:ETH:STR:SOUR:IP:RANG? 1 Returns: 1T128</pre>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:PMASK?

## SCPI Command Reference

MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:RANGe?**

<b>Description</b>	<p>This query returns the IPv4 multiplier range for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0To127.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon &gt; Test Configurator&gt; Set Up &gt;Streams &gt;IPv4 &gt; Source IP Multiplier</p> <p>Navigation Path: Test &gt; EtherSAM &gt; Test Configurator&gt; Set Up &gt;Services &gt;IPv4 &gt; Source IP Multiplier</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IP:RANGe? &lt;wsp&gt; &lt;Stream&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Range&gt;</p>
<b>Response(s)</b>	<p><b>Range:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the range for IPv4 multiplier.</p> <p>0T127, range from 1 to 128 is selected.</p> <p>1T128, range from 0 to 127 is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:SOUR:IP:MULT 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:SOUR:IP:RANG 1, 1T128</p> <p>SOUR:DATA:TEL:ETH:STR:SOUR:IP:RANG? 1</p> <p>Returns: 1T128</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:PMASK</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PAYLoad**

<b>Description</b>	<p>This command sets the payload pattern for the selected traffic stream.</p> <p>At *RST condition, this value is set to #HCC.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Streams &gt; MAC/IP/TCP &gt; Payload &gt;Pattern</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PAYLoad <wsp> <Tgen>, <Pattern>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Pattern:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the payload pattern value.</p> <p>Choices are #H00 through #HFF.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:PAYL 1, #HFF SOUR:DATA:TEL:ETH:STR:PAYL? 1 Returns: 255</pre>
<b>See Also</b>	SOURce[1..n]:DATA:TEL:ETH:STR:PROFile:RATE

---

## SCPI Command Reference

MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PAYLoad?**

<b>Description</b>	<p>This query returns payload pattern for the selected traffic stream.</p> <p>At *RST condition, this value is set to #HCC.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Streams &gt; MAC/IP/TCP &gt; Payload &gt;Pattern</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PAYLoad? &lt;wsp&gt; &lt;Tgen&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Pattern&gt;</p>
<b>Response(s)</b>	<p><b>Pattern:</b></p> <p>The response data syntax for this parameter is defined as a &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt; element.</p> <p>Return the payload value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:PAYL 1, #HFF</p> <p>SOUR:DATA:TEL:ETH:STR:PAYL? 1</p> <p>Returns: 255</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TEL:ETH:STR:PROFile:RATE?</p>



---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE**

---

**Description**

This command sets the type of Virtual Local Area Network (VLAN) identification for frame parameters specific VLAN service, direction, and stack.

At \*RST condition, this value is set to V8100.

Navigation Path: Test Configurator>Setup>Services>MAC/IP/UDP>Modify frame structure>VLAN>Type

**Syntax**

:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE <wsp>  
<Service>, LTOR | RTOL, <Stacked>, V8100 | V88A8 | V9100 | V9200 | V9300

---

#### :SOURce[1..n]:DATA:TELecom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE

Parameter(s)	Service:
	<p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p>
	<p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
	<p><b>Stacked:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p>
	<p><b>Idtype:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: V8100   V88A8   V9100   V9200   V9300</p> <p>Selects VLAN type.</p> <p>V8100: 8100</p> <p>V88A8: 88A8</p> <p>V9100: 9100</p> <p>V9200: 9200</p> <p>V9300: 9300</p>

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE**

---

**Example(s)**

SOURCE:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON

SOURCE:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC 1, LTOR,2

SOURCE:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE 1, LTOR,1,  
V88A8

SOURCE:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE? 1, LTOR,1

Returns: V88A8

**See Also**SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC

---

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE?**

<b>Description</b>	<p>This command sets the type of Virtual Local Area Network (VLAN) identification for frame parameters specific VLAN service, direction, and stack.</p> <p>At *RST condition, this value is set to V8100.</p> <p>Navigation Path: Test Configurator&gt;Setup&gt;Services&gt;MAC/IP/UDP&gt;Modify frame structure&gt;VLAN&gt;Type</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, &lt;Stacked&gt;</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Stacked:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p>
<b>Response Syntax</b>	<p>&lt;Idtype&gt;</p>

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE?**

---

<b>Response(s)</b>	<b>Idtype:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Virtual Local Area Network (VLAN) type. V8100, VLAN type 8100 is selected. V88A8, VLAN type 88A8 is selected. V9100, VLAN type 9100 is selected. V9200, VLAN type 9200 is selected. V9300, VLAN type 9300 is selected.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC 1, LTOR,2 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE 1, LTOR,1, V88A8 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE? 1, LTOR,1 Returns: V88A8
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC?

---

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID**

<b>Description</b>	<p>This command sets the Virtual Local Area Network (VLAN) Identification for frame of the selected service, direction, and stack.</p> <p>At *RST condition, this value is set to 2.</p> <p>Navigation Path: Test Configurator&gt;Setup&gt;Services&gt;MAC/IP/UDP&gt;Modify frame structure&gt;VLAN&gt;VLAN ID</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, &lt;Stacked&gt;, &lt;Id&gt;   MAXimum   MINimum</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID**

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Stacked:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p> <p><b>Id:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Select the VLAN ID.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC 1, LTOR,2</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID 1, LTOR,1, 50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:VLAN:TYPE</p>

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID?**

<b>Description</b>	<p>This query returns the available VLAN ID for frame of the selected service, direction, and stack.</p> <p>At *RST condition, this value is set to 2.</p> <p>Navigation Path: Test Configurator&gt;Setup&gt;Services&gt;MAC/IP/UDP&gt;Modify frame structure&gt;VLAN&gt;VLAN ID</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, &lt;Stacked&gt;[, MAXimum   MINimum]</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID?**

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Stacked:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p> <p>The program data syntax for the fourth parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Select the VLAN ID.</p> <p>This parameter is optional. If no token is specified, the current VLAN ID value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Id>

---

## SCPI Command Reference

MAC/IP/UDP

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID?**

<b>Response(s)</b>	<b>Id:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of VLAN identification.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC 1, LTOR,2 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID 1, LTOR,1, 50 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID? 1, LTOR,1 Returns: 50
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:TYPE?

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGiblebit**

---

<b>Description</b>	<p>This command enables or disables the user Virtual Local Area Network (VLAN) eligible bit for frame parameters for selected service, direction, and stack.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Configurator&gt;Setup&gt;Services&gt;MAC/IP/UDP&gt;Modify frame structure&gt;VLAN&gt;Drop Eligible</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGiblebit &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, &lt;Stacked&gt;, ON   OFF</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGiblebit**

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Stacked:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p> <p><b>Set:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable or disable the eligible bit for the specific VLAN Id.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC 1, LTOR,2</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGiblebit 1, LTOR,2,ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:VLAN:STAC</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGiblebit?**

<b>Description</b>	<p>This query returns the status of user VLAN eligible bit for frame parameters for selected service, direction, and stack.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Configurator&gt;Setup&gt;Services&gt;MAC/IP/UDP&gt;Modify frame structure&gt;VLAN&gt;Drop Eligible</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGiblebit? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, &lt;Stacked&gt;</code>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Stacked:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p>
<b>Response Syntax</b>	<code>&lt;Status&gt;</code>

---

## SCPI Command Reference

MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGiblebit?**

<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of eligible bit. 1, eligible bit is enabled. 0, eligible bit is disabled.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC 1, LTOR,2 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGiblebit 1, LTOR,2,ON SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:ID:ELIGiblebit? 1, LTOR,2 Returns: 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:VLAN:STAC?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRIority**

---

<b>Description</b>	<p>This command sets the priority of the Virtual Local Area Network (VLAN) identification for frame parameters for selected service, direction, and stack.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Configurator&gt;Setup&gt;Services&gt;MAC/IP/UDP&gt;Modify frame structure&gt;VLAN&gt;Priority</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRIority &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, &lt;Stacked&gt;, &lt;Priority&gt;   MAXimum   MINimum</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRIority**

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Stacked:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p> <p><b>Priority:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Set the priority of the stream.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC 1, LTOR,2</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRIority 1, LTOR,2,5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame</p>



**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRIority?**

<b>Description</b>	<p>This query returns the priority of the Virtual Local Area Network (VLAN) identification for frame parameters for selected service, direction, and stack.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Configurator&gt;Setup&gt;Services&gt;MAC/IP/UDP&gt;Modify frame structure&gt;VLAN&gt;Priority</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRIority? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, &lt;Stacked&gt;[, MAXimum   MINimum]</p>

---

## SCPI Command Reference

MAC/IP/UDP

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRIority?

<b>Parameter(s)</b>	<b>Service:</b> The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the Service number 1 or 10. <b>Direction:</b> The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are: LTOR   RTOL Selects the Direction. (For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction) LTOR: Local to Remote RTOL: Remote to Local <b>Stacked:</b> The program data syntax for the third parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the VLAN stacked.  The program data syntax for the fourth parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are: MAXimum   MINimum Set the priority of the stream. This parameter is optional. If no token is specified, the current priority of the Virtual Local Area Network (VLAN) value is returned. MAXimum: Biggest supported value MINimum: Smallest supported value
<b>Response Syntax</b>	<Priority>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRIority?**

<b>Response(s)</b>	<b>Priority:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns value of VLAN priority.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC 1, LTOR,2 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRIority 1, LTOR,2,5 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:PRIority? 1, LTOR,2 Returns: 5
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame?

---

## SCPI Command Reference

### MAC/IP/UDP

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:FLOoding**

<b>Description</b>	<p>This command enables the Destination MAC Flooding.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams&gt;MAC/IP/UDP &gt; MAC &gt; Destination Flooding</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:FLOoding &lt;wsp&gt; &lt;Stream&gt;, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Destination MAC Flooding.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:ADDR:DEST:FLO 1,ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:FLOoding?**

<b>Description</b>	<p>This query returns status of Destination MAC Flooding.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams&gt;MAC/IP/UDP &gt; MAC &gt; Destination Flooding</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:FLOoding? &lt;wsp&gt; &lt;Stream&gt;</pre>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<pre>&lt;Set&gt;</pre>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Destination MAC Flooding.</p> <p>1, Destination MAC Flooding is enabled.</p> <p>0, Destination MAC Flooding is disabled.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:ADDR:DEST:FLO 1,ON SOUR:DATA:TEL:ETH:STR:ADDR:DEST:FLO? 1 Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce</pre>

---

## SCPI Command Reference

### MAC/IP/UDP

---

<b>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:FLOoding:RANGe</b>	
<b>Description</b>	<p>This command sets the Source/Destination MAC Flooding range.</p> <p>At *RST condition, this value is set to 4096.</p> <p>Navigation Path: Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams&gt;MAC/IP/UDP &gt; MAC &gt; Flooding Range</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:FLOoding:RANGe &lt;wsp&gt; &lt;Stream&gt;, &lt;Range&gt;</pre>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Range:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Sets the Source/Destination MAC Flooding range.</p> <p>Range is from 2 to 4096.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:ADDR:FLO:RANG 1,64</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination?</pre>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:FLOoding:RANGe?**

<b>Description</b>	<p>This query returns the Source/Destination MAC Flooding range.</p> <p>At *RST condition, this value is set to 4096.</p> <p>Navigation Path: Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams&gt;MAC/IP/UDP &gt; MAC &gt; Flooding Range</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:FLOoding:RANGe? &lt;wsp&gt;&lt;Stream&gt;</code>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<code>&lt;Set&gt;</code>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Source/Destination MAC Flooding range.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:ADDR:FLO:RANG 1,64 SOUR:DATA:TEL:ETH:STR:ADDR:FLO:RANG? 1 Returns: 64</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination</code>

---

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:FLOoding**

---

<b>Description</b>	<p>This command enables the Source MAC Flooding.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams&gt;MAC/IP/UDP &gt; MAC &gt; Source Flooding</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:FLOoding &lt;wsp&gt; &lt;Stream&gt;, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Source MAC Flooding.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:ADDR:SOUR:FLO 1, ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:IP?</p>

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:FLOoding?**

<b>Description</b>	<p>This query returns Source MAC Flooding.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams&gt;MAC/IP/UDP &gt; MAC &gt; Source Flooding</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:FLOoding? &lt;wsp&gt; &lt;Stream&gt;</code>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<code>&lt;Set&gt;</code>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Source MAC Flooding.</p> <p>1, Source MAC Flooding is enabled.</p> <p>0, Source MAC Flooding is disabled.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:ADDR:SOUR:FLO 1,ON SOUR:DATA:TEL:ETH:STR:ADDR:SOUR:FLO? 1 Returns: 1</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:SOURce:IP</code>

---

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:MULTiplicat

---

<b>Description</b>	<p>This command enables or disables the IPv6 multiplier for the selected traffic stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt;IPv6 &gt; Source IP Multiplier</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:MULTiplicat <wsp> <Stream>, ON   OFF
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the IPv6 multiplier for the selected traffic stream.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:SOUR:IPV:MULT 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:SOUR:IPV:MULT? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:IPv6:HOP:LIMit

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:MULTiplicat?**

<b>Description</b>	<p>This query returns the status of IPv6 multiplicator for the selected traffic stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt;IPv6 &gt; Source IP Multiplicator</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:MULTiplicat? &lt;wsp&gt; &lt;Stream&gt;</pre>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<pre>&lt;Status&gt;</pre>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of IPv6 multiplicator.</p> <p>1, IPv6 multiplicator is enabled.</p> <p>0, IPv6 multiplicator is disabled.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:SOUR:IPV:MULT 1, ON SOUR:DATA:TEL:ETH:STR:SOUR:IPV:MULT? 1 Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:IPv6:HOP:LIMit?</pre>

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:RANGe**

<b>Description</b>	<p>This command sets the IPv6 multiplicator range for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1T0128.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Streams &gt;MAC/IP/UDP&gt;IPv6 &gt; Source IP Multiplicator</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:RANGe &lt;wsp&gt; &lt;Stream&gt;, 0T127   1T128</pre>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Range:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 0T127   1T128</p> <p>Sets the range for IPv6 multiplicator.</p> <p>0T127: the range from 0T127.</p> <p>1T128: the range from 1T128.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:SOUR:IPV:MULT 1, ON SOUR:DATA:TEL:ETH:STR:SOUR:IPV:RANG 1, 1T128 SOUR:DATA:TEL:ETH:STR:SOUR:IPV:RANG? 1 Returns: 1T128</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:PMASK? SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:MODE</pre>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:RANGe?**

<b>Description</b>	<p>This query returns the IPv6 multiplier range for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1T128.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Streams &gt;IPv6 &gt; Source IP Multiplier</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:SOURce:IPVersion:RANGe? <wsp> <Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Range>
<b>Response(s)</b>	<p><b>Range:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the range for IPv6 multiplier.</p> <p>1T128, range from 1T128 is selected.</p> <p>0T127, range from 0T127 is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:SOUR:IPV:MULT 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:SOUR:IPV:RANG 1, 1T128</p> <p>SOUR:DATA:TEL:ETH:STR:SOUR:IPV:RANG? 1</p> <p>Returns: 1T128</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:PMASK</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:MODE</p>

---

## SCPI Command Reference

### MAC/IP/UDP

---

<b>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IPVersion</b>	
<b>Description</b>	<p>This command sets the Destination IPv6 address.</p> <p>At *RST condition, this value is set to ""0000:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt;IPv6 &gt;IPv6 Destination Address</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IPVersion &lt;wsp&gt; &lt;Stream&gt;, &lt;Address&gt;</pre>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Destination IPv6 address.</p>
<b>Example(s)</b>	<pre>SENSe:DATA:TELEcom:ETHernet:STReam:DESTination:IPV 1, "FE80:0000:0000:0000:0200:00FF:FE00:0000" SENSe:DATA:TELEcom:ETHernet:STReam:DESTination:IPV? 1 Returns: "0000:0000:0000:0000:0000:0000:0000:0000"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPV?</pre>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IPVersion?**

<b>Description</b>	<p>This query returns the Destination IPv6 address.</p> <p>At *RST condition, this value is set to ""0000:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test&gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt;IPv6 &gt;IPv6 Destination Address</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:DESTination:IPVersion? <wsp> <Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Destination IPv6 address.</p>
<b>Example(s)</b>	<p>SENSe:DATA:TELEcom:ETHernet:STReam:DESTination:IPV 1, "FE80:0000:0000:0000:0200:00FF:FE00:0000"</p> <p>SENSe:DATA:TELEcom:ETHernet:STReam:DESTination:IPV? 1</p> <p>Returns: "0000:0000:0000:0000:0000:0000:0000:0000"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPV

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FLABel:IPVersion

---

<b>Description</b>	<p>This command sets the IPv6 Flow Label.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt; IPv6 &gt; Flow Label.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FLABel:IPVersion <wsp> <Stream>, <Flabel>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Flabel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the IPv6 Flow Label.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FLAB:IPV 1, 20</p> <p>SENS:DATA:TEL:ETH:STR:FLAB:IPV? 1</p> <p>Returns: 20</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IPV?

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FLABel:IPVersion?**

<b>Description</b>	<p>This query returns the IPv6 Flow Label.</p> <p>At *RST condition, this value is set to 0</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt; IPv6 &gt; Flow Label.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FLABel:IPVersion? <wsp> <Stream>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional.</p> <p>If no token is specified, the current flow label is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Flabel>
<b>Response(s)</b>	<p><b>Flabel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the IPv6 Flow Label.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FLAB:IPV 1, 20</p> <p>SENS:DATA:TEL:ETH:STR:FLAB:IPV? 1</p> <p>Returns: 21</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IPV

## SCPI Command Reference

### MAC/IP/UDP

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IPVersion:HOP:LIMit

<b>Description</b>	<p>This command Set The Hop Limit field of the IPv6 header shall be configurable.</p> <p>At *RST condition, this value is set to</p> <p>Range = 1 to 255</p> <p>Resolution = 1</p> <p>Default = 128</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Streams &gt; MAC/IP/UDP &gt; IPv6 &gt; Hop Limit</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IPVersion:HOP:LIMit &lt;wsp&gt; &lt;Stream&gt;, &lt;HOP Limit&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>HOP Limit:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the HOP Limit</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:STReam:IPV:HOP:LIMit 1, 25</p> <p>SOURce:DATA:TELEcom:ETHernet:STReam:IPV:HOP:LIMit? 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:ADDRESS?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:IICoupled</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IPVersion:HOP:LIMit?**

<b>Description</b>	<p>This query returns The Hop Limit field of the IPv6 header shall be configurable.</p> <p>At *RST condition, this value is set to</p> <p>Range = 1 to 255</p> <p>Resolution = 1</p> <p>Default = 128</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Streams &gt; MAC/IP/UDP &gt; IPv6 &gt; Hop Limit</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IPVersion:HOP:LIMit? &lt;wsp&gt; &lt;Stream&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current Hop Limit is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;HOP Limit&gt;</p>

## SCPI Command Reference

MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:IPVersion:HOP:LIMit?**

<b>Response(s)</b>	<b>HOP Limit:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the IPv6 Hop Limit.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:STReam:IPV:HOP:LIMit 1, 25 SOURce:DATA:TELEcom:ETHernet:STReam:IPV:HOP:LIMit? 2
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:ADDRess SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:MODE SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:IICoupled

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MAC:OUI**

<b>Description</b>	<p>This command sets the frame format OUI type</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen &amp; Mon &gt; Setup &gt; Stream &gt; MAC/IP/UDP &gt; MAC &gt; OUI</p> <p>Available for Frame Format - 802.3 SNAP only</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MAC:OUI <wsp>RFC1042   8021H   USERDEFINED
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RFC1042   8021H   USERDEFINED</p> <p>Sets Frame format type</p> <p>RFC1042, sets frame format type to RFC1042</p> <p>8021H,sets frame format type to 8021H</p> <p>USERDEFINED, sets frame format type to USERDEFINED</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MAC:OUI RFC1042</p> <p>SOUR:DATA:TEL:ETH:STR:MAC:OUI?</p> <p>Returns: RFC1042</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:NETWork:FRAMe:FORMat:OUI RFC1042</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:NETWork:FRAMe:FORMat:OUI?</p> <p>Returns RFC1042</p>

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MAC:OUI?

---

<b>Description</b>	<p>This query returns the frame format OUI type</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen &amp; Mon &gt; Setup &gt; Stream &gt; MAC/IP/UDP &gt; MAC &gt; OUI</p> <p>Available for Frame Format - 802.3 SNAP only</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MAC:OUI?
<b>Response Syntax</b>	<OUI>
<b>Response(s)</b>	<p><b>OUI:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns Frame format type</p> <p>RFC1042, Returns as RFC1042 frame format.</p> <p>8021H,Returns 8021H as frame format.</p> <p>USERDEFINED,Returns USERDEFINED as frame format.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MAC:OUI RFC1042</p> <p>SOUR:DATA:TEL:ETH:STR:MAC:OUI?</p> <p>Returns: RFC1043</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:NETWork:FRAMe:FORMat:OUI RFC1042</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:NETWork:FRAMe:FORMat:OUI?</p> <p>Returns RFC1042</p>

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:LABel**

---

<b>Description</b>	<p>This command sets the Multi Protocol Label Switching (MPLS) label type.</p> <p>At *RST condition, this value is set to 16.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt; MPLS &gt;Label</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:LABel &lt;wsp&gt; &lt;Stream&gt;, &lt;Label Index&gt;, &lt;Label Value&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the stream from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Label Index:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Label Index.</p> <p><b>Label Value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the label value.</p> <p>Choices are 0 through 1048575.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:MPLS:LAB 1, 1, 16 SOUR:DATA:TEL:ETH:STR:MPLS:LAB? 1, 1 Returns: 16</pre>
<b>See Also</b>	<code>SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL</code>

---

## SCPI Command Reference

### MAC/IP/UDP

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:LABel?

---

<b>Description</b>	<p>This query returns the Multi Protocol Label Switching (MPLS) label type.</p> <p>At *RST condition, this value is set to 16.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt; MPLS &gt;Label</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:LABel? &lt;wsp&gt; &lt;Stream&gt;, &lt;Label Index&gt;[, MAXimum   MINimum ]</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the stream from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Label Index:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Label Index.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current label number is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Label&gt;</p>

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:LABel?**

<b>Response(s)</b>	<b>Label:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the label value.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:MPLS:LAB 1, 1, 16 SOUR:DATA:TEL:ETH:STR:MPLS:LAB? 1, 1 Returns: 16
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:MPLS:HEADers SOURce[1..n]:DATA:TELEcom:ETHernet:MPLS:HEADers?

---

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSExp**

<b>Description</b>	<p>This command selects the COS/EXP value.</p> <p>At *RST condition, this value is set to 0 (000 - Low) (default).</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Test Configurator &gt; Set Up &gt; Streams &gt; MAC/IP/UDP &gt; MPLS &gt; Label &gt; COS</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSExp &lt;wsp&gt; &lt;Stream&gt;, &lt;Label Index&gt;, 000LOW   001LOW   010LOW   011LOW   100HIGH   101HIGH   110HIGH   111HIGH</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSExp**

<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the stream from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Label Index:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Label Index.</p> <p><b>Cosexp:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 000LOW   001LOW   010LOW   011LOW   100HIGH   101HIGH   110HIGH   111HIGH</p> <p>Selects the COS/EXP value.</p> <p>Sets the Class Of Service for the instrument.</p> <p>000LOW: 0 (000 - Low)</p> <p>001LOW: 1 (001 - Low)</p> <p>010LOW: 2 (010 - Low)</p> <p>011LOW: 3 (011 - Low)</p> <p>100HIGH: 4 (100 - High)</p> <p>101HIGH: 5 (101 - High)</p> <p>110HIGH: 6 (110 - High)</p> <p>111HIGH: 7 (111 - High)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MPLS:COS 1, 1, 111HIGH</p> <p>SOUR:DATA:TEL:ETH:STR:MPLS:COS? 1, 1</p> <p>Returns: 111HIGH</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:MPLS:HEADers</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:MPLS:HEADers?</p>

---

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSExp?**

<b>Description</b>	<p>This query returns the COS/EXP value.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt; MPLS &gt;Label &gt;COS</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSExp? &lt;wsp&gt; &lt;Stream&gt;, &lt;Label Index&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the stream from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Label Index:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Label Index.</p>
<b>Response Syntax</b>	<p>&lt;Cosexp&gt;</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSeXp?**

<b>Response(s)</b>	<b>Cosexp:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns Class Of Service for the instrument. 000LOW, returns 0 (000 - Low) as Class Of Service. 001LOW, returns 1 (001 - Low) as Class Of Service. 010LOW, returns 2 (010 - Low) as Class Of Service. 011LOW, returns 3 (011 - Low) as Class Of Service. 100HIGH, returns 4 (100 - High) as Class Of Service. 101HIGH, returns 5 (101 - High) as Class Of Service. 110HIGH, returns 6 (110 - High) as Class Of Service. 111HIGH, returns 7 (111 - High) as Class Of Service.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:MPLS:COS 1, 1, 000LOW SOUR:DATA:TEL:ETH:STR:MPLS:COS? 1, 1 Returns: 000LOW
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS

---

## SCPI Command Reference

### MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL**

<b>Description</b>	<p>This command sets the Time to Live (TTL) value of Multi Protocol Label Switching (MPLS). At *RST condition, this value is set to 128.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt; MPLS &gt;Label &gt;TTL</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL &lt;wsp&gt; &lt;Stream&gt;, &lt;Label Index&gt;, &lt;TTL&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the stream from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Label Index:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Label Index.</p> <p><b>TTL:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the TTL value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MPLS:TTL 1, 1, 200</p> <p>SOUR:DATA:TEL:ETH:STR:MPLS:TTL? 1, 1</p> <p>Returns: 200</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSExp</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL?**

<b>Description</b>	<p>This query returns the Time to Live (TTL) value of Multi Protocol Label Switching (MPLS). At *RST condition, this value is set to 128.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt; MPLS &gt;Label &gt;TTL</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL? <wsp> <Stream>, <Label Index>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the stream from 1 to 10. Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Label Index:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element. Selects the Label Index.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current TTL value is returned. MAXimum: Biggest supported value MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<TTL>

## SCPI Command Reference

MAC/IP/UDP

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL?**

<b>Response(s)</b>	<b>TTL:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the TTL value.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:MPLS:TTL 1, 1, 200 SOUR:DATA:TEL:ETH:STR:MPLS:TTL? 1, 1 Returns: 200
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ELECtrical:STReam:MPLS:LABel?

---



# EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RATE**

---

<b>Description</b>	<p>This command sets the transmitter payload rate for the selected traffic payload type.</p> <p>At *RST condition, this value is set to 100%.</p> <p>Navigation Path: EtherBERT &gt; Test Setup &gt; Test Configurator &gt; Shaping &gt; TX Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RATE <wsp> <Rate>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects transmitter payload rate for the selected traffic payload type.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:RATE 56666</p> <p>SOUR:DATA:TEL:ETH:STR:RATE?</p> <p>Returns: 56666</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RATE?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame</p>

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RATE?**

<b>Description</b>	<p>This query returns the transmitter payload rate for the selected traffic payload type.</p> <p>At *RST condition, this value is set to 100%.</p> <p>Navigation Path: EtherBERT &gt; Test Setup &gt; Test Configurator &gt; Shaping &gt; TX Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RATE? [ &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum ]</p>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current transmitter payload rate is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the stream rate for the transmitter.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:RATE 56666</p> <p>SOUR:DATA:TEL:ETH:STR:RATE?</p> <p>Returns: 56666</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RATE</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TX:STATus**

---

<b>Description</b>	<p>This command sets the transmitter status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: EtherBERT &gt; Test Setup &gt; Test Configurator &gt; Shaping &gt; Enable TX</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TX:STATus <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the transmitter status.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TX:STAT ON</p> <p>SOUR:DATA:TEL:ETH:STR:TX:STAT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELcom:ETHernet:STReam:TX:STATus

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TX:STATus?**

---

<b>Description</b>	This query returns the transmitter status. At *RST condition, this value is set to OFF. Navigation Path: EtherBERT > Test Setup > Test Configurator > Shaping > Enable TX
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TX:STATus?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<b>Set:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the transmitter status. ON, sets transmitter status to ON. OFF, sets transmitter status to OFF.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:TX:STAT ON SOUR:DATA:TEL:ETH:STR:TX:STAT? Returns: 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TX:STATus?

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE**

<b>Description</b>	<p>This command sets the test pattern type for the transmitter.</p> <p>At *RST condition, this value is set to PRBs2E31.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Tx Pattern</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE &lt;wsp&gt;PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31   UPATTERN   NCLIENT   PRBS31UNSCRAMBLED   SEEDA   SEEDB</p>
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31   UPATTERN   NCLIENT   PRBS31UNSCRAMBLED   SEEDA   SEEDB</p> <p>Selects the pattern type for the transmitter.</p> <p>PRBs2E9: PRBS9</p> <p>PRBs2E11: PRBS11</p> <p>PRBs2E15: PRBS15</p> <p>PRBs2E20: PRBS20</p> <p>PRBs2E23: PRBS23</p> <p>PRBs2E31: PRBS31</p> <p>UPATTERN: User Defined</p> <p>NCLIENT: NULL CLIENT</p> <p>PRBS31UNSCRAMBLED: PRBS31 Unscrambled</p> <p>SEEDA: Seed A</p> <p>SEEDB: Seed B</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:TYPE PRBs2E15</p> <p>SOUR:DATA:TEL:PATT:TYPE?</p> <p>Returns: PRBs2E15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATtern:TYPE?</p>

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE?**

<b>Description</b>	<p>This query returns the test pattern type for transmitter.</p> <p>At *RST condition, this value is set to PRBs2E31.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Tx Pattern</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE?
<b>Response Syntax</b>	<Pattern>
<b>Response(s)</b>	<p><b>Pattern:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the pattern type for the transmitter.</p> <p>PRBs2E9, pattern type PRBS9 is selected.</p> <p>PRBs2E11, pattern type PRBS11 is selected.</p> <p>PRBs2E15, pattern type PRBS15 is selected.</p> <p>PRBs2E20, pattern type PRBS20 is selected.</p> <p>PRBs2E23, pattern type PRBS23 is selected.</p> <p>PRBs2E31, pattern type PRBS31 is selected.</p> <p>UPATTERN, pattern type User Defined is selected.</p> <p>NCLIENT, pattern type NCLIENT is selected.</p> <p>PRBS31UNSCRAMBLED, pattern type PRBS31 Unscrambled is selected.</p> <p>SEEDA, pattern type Seed A is selected.</p> <p>SEEDB, pattern type Seed B is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:TYPE PRBs2E15</p> <p>SOUR:DATA:TEL:PATT:TYPE?</p> <p>Returns: PRBs2E15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:TYPE

---

**:SENSe[1..n]:DATA:TELecom:PATtern:TYPE**

---

<b>Description</b>	<p>This command selects test pattern type for the receiver.</p> <p>At *RST condition, this value is set to PRBs2E31.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Rx Pattern</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELecom:PATtern:TYPE &lt;wsp&gt;PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31   UPATTERN   NCLIENT   PRBS31UNSCRAMBLED   SEEDA   SEEDB</p>
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31   UPATTERN   NCLIENT   PRBS31UNSCRAMBLED   SEEDA   SEEDB</p> <p>Selects the pattern type for the receiver.</p> <p>PRBs2E9: PRBS9.</p> <p>PRBs2E11: PRBS11.</p> <p>PRBs2E15: PRBS15.</p> <p>PRBs2E20: PRBS20.</p> <p>PRBs2E23: PRBS23.</p> <p>PRBs2E31: PRBS31.</p> <p>UPATTERN: User Defined.</p> <p>NCLIENT: NULL CLIENT.</p> <p>PRBS31UNSCRAMBLED: PRBS31 Unscrambled.</p> <p>SEEDA: Seed A.</p> <p>SEEDB: Seed B.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:PATT:TYPE PRBs2E15</p> <p>SENS:DATA:TEL:PATT:TYPE?</p> <p>Returns: PRBs2E15</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:PATtern:TYPE?</p>

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE?**

<b>Description</b>	<p>This query returns test pattern type for the receiver.</p> <p>At *RST condition, this value is set to PRBs2E31.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Rx Pattern</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE?
<b>Response Syntax</b>	<Pattern>
<b>Response(s)</b>	<p><b>Pattern:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the pattern type for the receiver.</p> <p>PRBs2E9, pattern type PRBS9 is selected.</p> <p>PRBs2E11, pattern type PRBS11 is selected.</p> <p>PRBs2E15, pattern type PRBS15 is selected.</p> <p>PRBs2E20, pattern type PRBS20 is selected.</p> <p>PRBs2E23, pattern type PRBS23 is selected.</p> <p>PRBs2E31, pattern type PRBS31 is selected.</p> <p>UPATTERN, pattern type User Defined is selected.</p> <p>NCLIENT, pattern type NCLIENT is selected.</p> <p>PRBS31UNSCRAMBLED, pattern type PRBS31 Unscrambled is selected.</p> <p>SEEDA, pattern type Seed A is selected.</p> <p>SEEDB, pattern type Seed B is selected.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:PATT:TYPE PRBs2E15</p> <p>SENS:DATA:TEL:PATT:TYPE?</p> <p>Returns: PRBs2E15</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:PATtern:TYPE

---



---

**:SOURce[1..n]:DATA:TELEcom:POLarity**

---

<b>Description</b>	<p>This command sets the polarity pattern for the transmitter.</p> <p>At *RST condition, this value is set to NINVerted.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Tx Pattern &gt; Invert</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:POLarity <wsp>NINVerted   INVerted
<b>Parameter(s)</b>	<p><b>Polarity:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NINVerted   INVerted</p> <p>Selects the transmitter polarity.</p> <p>NINVerted: Non-inverted.</p> <p>INVerted: Inverted.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:POL INV</p> <p>SOUR:DATA:TEL:POL?</p> <p>Returns: INVerted</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATTern:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:POLarity?</p>

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SOURce[1..n]:DATA:TELEcom:POLarity?**

<b>Description</b>	<p>This query returns the polarity pattern of the transmitter.</p> <p>At *RST condition, this value is set to NINVerted.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Tx Pattern &gt; Invert</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:POLarity?
<b>Response Syntax</b>	<Polarity>
<b>Response(s)</b>	<p><b>Polarity:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the transmitter polarity pattern for specific stream.</p> <p>NINVerted, selects the polarity as non-inverted.</p> <p>INVerted, selects the polarity as inverted.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:POL INV</p> <p>SOUR:DATA:TEL:POL?</p> <p>Returns: INVerted</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATTern:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:PATTern:POLarity</p>

---

**:SENSe[1..n]:DATA:TELEcom:POLarity**

<b>Description</b>	<p>This command sets the polarity pattern of the receiver.</p> <p>At *RST condition, this value is set to NINVerted.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Rx Pattern &gt; Invert</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:POLarity <wsp>NINVerted   INVerted
<b>Parameter(s)</b>	<p><b>Polarity:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NINVerted   INVerted</p> <p>Selects the polarity pattern for the receiver.</p> <p>NINVerted: Non-inverted.</p> <p>INVerted: Inverted.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:POL INV</p> <p>SENS:DATA:TEL:POL?</p> <p>Returns: INVerted</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:PATTern:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:PATTern:POLarity?</p>

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SENSe[1..n]:DATA:TELEcom:POLarity?**

<b>Description</b>	<p>This query returns the polarity pattern of the receiver.</p> <p>At *RST condition, this value is set to NINVerted.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Rx Pattern &gt; Invert</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:POLarity?
<b>Response Syntax</b>	<Polarity>
<b>Response(s)</b>	<p><b>Polarity:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the polarity pattern of the receiver.</p> <p>NINVERTED, priority as non-inverted is selected.</p> <p>INVERTED, priority as inverted is selected.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:POL INV</p> <p>SENS:DATA:TEL:POL?</p> <p>Returns: INVerted</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:PATtern:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:PATtern:POLarity</p>

---

**:SENSe[1..n]:DATA:TELEcom:COUPled**

<b>Description</b>	<p>This command enables or disables the receiver coupling.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Coupled Rx to Tx</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:COUPled <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the receiver coupling.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:COUP ON</p> <p>SENS:DATA:TEL:COUP?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:COUPled?

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SENSe[1..n]:DATA:TELEcom:COUPled?**

<b>Description</b>	<p>This query returns the status of the receiver coupling.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Coupled Rx to Tx</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:COUPled?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the coupled status of the receiver.</p> <p>ON, sets the receiver status to ON.</p> <p>OFF, sets the receiver status to OFF.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:COUP ON</p> <p>SENS:DATA:TEL:COUP?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:COUPled

---

---

**:SENSe[1..n]:DATA:TELEcom:PATTern:RXPanalysis:STATus**

---

<b>Description</b>	<p>This command sets the status of the RX Pattern Analysis.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; RX Pattern Analysis</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:PATTern:RXPanalysis:STATus <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status of the RX Pattern Analysis.</p>
<b>Example(s)</b>	SENS:DATA:TEL:PATT:RXP:STAT ON
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:COUPled

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

---

### :SENSe[1..n]:DATA:TELEcom:PATtern:RXPanalysis:STATus?

---

<b>Description</b>	<p>This query returns the status of the RX Pattern Analysis.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; RX Pattern Analysis</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:PATtern:RXPanalysis:STATus?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the pattern analysis status of the receiver.</p> <p>ON, sets RX Pattern Analysis status to ON.</p> <p>OFF, sets RX Pattern Analysis status to OFF.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:PATT:RXP:STAT ON</p> <p>SENS:DATA:TEL:PATT:RXP:STAT?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:COUPled?

---



**:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue**

<b>Description</b>	<p>This command sets the transmitter user pattern value for the specified index.</p> <p>At *RST condition, this value is set to #H00000000.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Tx &gt; User Pattern</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue <wsp> <Value>
<b>Parameter(s)</b>	<p><b>Value:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the transmitter user pattern value for the specified index.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:TYPE:USER:VAL #H00000001</p> <p>SOUR:DATA:TEL:PATT:TYPE:USER:VAL?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATtern:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?</p>

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?**

<b>Description</b>	<p>This query returns the transmitter user pattern value for the specified index.</p> <p>At *RST condition, this value is set to #H00000000.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Tx &gt; User Pattern</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the user pattern value for the transmitter.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:TYPE:USER:VAL #H00000001</p> <p>SOUR:DATA:TEL:PATT:TYPE:USER:VAL?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATtern:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue</p>

---

**:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue**

<b>Description</b>	<p>This command sets the receiver user pattern value for the specified index.</p> <p>At *RST condition, this value is set to #H00000000.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Rx &gt; User Pattern</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue <wsp> <Value>
<b>Parameter(s)</b>	<p><b>Value:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the receiver user pattern value for the specified index.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:PATT:TYPE:USER:VAL #H00000001</p> <p>SENS:DATA:TEL:PATT:TYPE:USER:VAL?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:PATtern:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?</p>

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

---

### :SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?

---

<b>Description</b>	<p>This query returns the receiver user pattern value for the specified index.</p> <p>At *RST condition, this value is set to #H00000000.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Pattern &gt; Rx &gt; User Pattern</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the user pattern value for the receiver.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:PATT:TYPE:USER:VAL #H00000001</p> <p>SENS:DATA:TEL:PATT:TYPE:USER:VAL?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:PATtern:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL**

---

<b>Description</b>	<p>This command selects all lanes for further configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; All Lanes</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Selects all receiver lanes.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:UPRB:PATT:GLOB:ALL OFF</p> <p>SENS:DATA:TEL:UPRB:PATT:GLOB:ALL?</p> <p>Returns: 0</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:COUPled?

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL?**

<b>Description</b>	This query returns the status of all lanes for further configuration. At *RST condition, this value is set to NINVerted. Navigation Path: BERT > Test Setup > Unframed BERT > Pattern > All Lanes
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<b>Set:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of all lanes of the receiver. 1, All lanes of the receiver status is enabled. 0, All lanes of the receiver status is disabled.
<b>Example(s)</b>	SENS:DATA:TEL:UPRB:PATT:GLOB:ALL ON SENS:DATA:TEL:UPRB:PATT:GLOB:ALL? Returns: 1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:COUPled

---

---

**:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:TX**

---

<b>Description</b>	<p>This command sets the global test pattern type for the TRANSMITTER.</p> <p>At *RST condition, this value is set to PRBs2E23.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; TX</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:TX          &lt;wsp&gt;PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31   PSWAVE1            PSWAVE2   PSWAVE4   PSWAVE8   PSWAVE16</p>
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31   PSWAVE1   PSWAVE2   PSWAVE4   PSWAVE8   PSWAVE16</p> <p>Selects the test pattern for the Transmitter.</p> <p>PRBs2E9, pattern type is PRBS 2<sup>9</sup>-1.</p> <p>PRBs2E11, pattern type is PRBS 2<sup>11</sup>-1.</p> <p>PRBs2E15, pattern type is PRBS 2<sup>15</sup>-1.</p> <p>PRBs2E20, pattern type is PRBS 2<sup>20</sup>-1.</p> <p>PRBs2E23, pattern type is PRBS 2<sup>23</sup>-1.</p> <p>PRBs2E31, pattern type is PRBS 2<sup>31</sup>-1.</p> <p>PSWAVE1, pattern type is PSWAVE1.</p> <p>PSWAVE2, pattern type is PSWAVE2.</p> <p>PSWAVE4, pattern type is PSWAVE4.</p> <p>PSWAVE8, pattern type is PSWAVE8.</p> <p>PSWAVE16, pattern type is PSWAVE16.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UPRB:PATT:GLOB:PRBS:TYPE:TX PRBs2E23</p> <p>SOUR:DATA:TEL:UPRB:PATT:GLOB:PRBS:TYPE:TX?</p> <p>Returns: PRBs2E23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX</p>

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPR1), and Unframed BERT*

---

**:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:TX?**

<b>Description</b>	This query returns the global test pattern type for the TRANSMITTER. At *RST condition, this value is set to PRBs2E23. Navigation Path: BERT > Test Setup > Unframed BERT > Pattern > TX
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:TX?
<b>Response Syntax</b>	<Pattern>
<b>Response(s)</b>	<b>Pattern:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the test pattern for TRANSMITTER PRBs2E9, pattern type PRBS 2 <sup>9</sup> -1 is selected. PRBs2E11, pattern type PRBS 2 <sup>11</sup> -1 is selected. PRBs2E15, pattern type PRBS 2 <sup>15</sup> -1 is selected. PRBs2E20, pattern type PRBS 2 <sup>20</sup> -1 is selected. PRBs2E23, pattern type PRBS 2 <sup>23</sup> -1 is selected. PRBs2E31, pattern type PRBS 2 <sup>31</sup> -1 is selected. PSWAVE1, pattern type PSWAVE1 is selected. PSWAVE2, pattern type PSWAVE2 is selected. PSWAVE4, pattern type PSWAVE4 is selected. PSWAVE8, pattern type PSWAVE8 is selected. PSWAVE16, pattern type PSWAVE16 is selected.
<b>Example(s)</b>	SOUR:DATA:TEL:UPRB:PATT:GLOB:PRBS:TYPE:TX?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX?

---



**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX**

<b>Description</b>	<p>This command sets the global test pattern type for the receiver.</p> <p>At *RST condition, this value is set to PRBs2E23.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; RX</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX          &lt;wsp&gt;PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31   PSWAVE1            PSWAVE2   PSWAVE4   PSWAVE8   PSWAVE16</p>
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31   PSWAVE1   PSWAVE2   PSWAVE4   PSWAVE8   PSWAVE16</p> <p>Selects the test pattern for the Receiver.</p> <p>PRBs2E9, pattern type is PRBS 2<sup>9</sup>-1.</p> <p>PRBs2E11, pattern type is PRBS 2<sup>11</sup>-1.</p> <p>PRBs2E15, pattern type is PRBS 2<sup>15</sup>-1.</p> <p>PRBs2E20, pattern type is PRBS 2<sup>20</sup>-1.</p> <p>PRBs2E23, pattern type is PRBS 2<sup>23</sup>-1.</p> <p>PRBs2E31, pattern type is PRBS 2<sup>31</sup>-1.</p> <p>PSWAVE1, pattern type is PSWAVE1.</p> <p>PSWAVE2, pattern type is PSWAVE2.</p> <p>PSWAVE4, pattern type is PSWAVE4.</p> <p>PSWAVE8, pattern type is PSWAVE8.</p> <p>PSWAVE16, pattern type is PSWAVE16.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:UPRB:PATT:GLOB:PRBS:TYPE:RX PRBs2E31</p> <p>SENS:DATA:TEL:UPRB:PATT:GLOB:PRBS:TYPE:RX?</p> <p>Returns: PRBs2E31</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX?</p>

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPR1), and Unframed BERT*

---

**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX?**

<b>Description</b>	<p>This query returns the global test pattern type for the receiver.</p> <p>At *RST condition, this value is set to PRBs2E23.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; RX</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX?
<b>Response Syntax</b>	<Pattern>
<b>Response(s)</b>	<p><b>Pattern:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the test pattern for Receiver.</p> <p>PRBs2E9, pattern type PRBS 2<sup>9</sup>-1 is selected.</p> <p>PRBs2E11, pattern type PRBS 2<sup>11</sup>-1 is selected.</p> <p>PRBs2E15, pattern type PRBS 2<sup>15</sup>-1 is selected.</p> <p>PRBs2E20, pattern type PRBS 2<sup>20</sup>-1 is selected.</p> <p>PRBs2E23, pattern type PRBS 2<sup>23</sup>-1 is selected.</p> <p>PRBs2E31, pattern type PRBS 2<sup>31</sup>-1 is selected.</p> <p>PSWAVE1, pattern type PSWAVE1 is selected.</p> <p>PSWAVE2, pattern type PSWAVE2 is selected.</p> <p>PSWAVE4, pattern type PSWAVE4 is selected.</p> <p>PSWAVE8, pattern type PSWAVE8 is selected.</p> <p>PSWAVE16, pattern type PSWAVE16 is selected.</p>
<b>Example(s)</b>	SENS:DATA:TEL:UPRB:PATT:GLOB:PRBS:TYPE:RX?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX

---

**:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:TX**

<b>Description</b>	<p>This command sets the global polarity pattern for the transmitter.</p> <p>At *RST condition, this value is set to NINVerted.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; TX</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:TX <wsp>NINVerted   INVerted
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NINVerted   INVerted</p> <p>Selects the polarity for the transmitter.</p> <p>NINVerted, the polarity selected is Non-Inverted.</p> <p>INVerted, the polarity selected is Inverted.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UPRB:PATT:GLOB:POL:TX INVerted</p> <p>SOUR:DATA:TEL:UPRB:PATT:GLOB:POL:TX?</p> <p>Returns: INVerted</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATtern:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:POLarity?</p>

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:TX?**

<b>Description</b>	This query returns the global polarity pattern for the transmitter. At *RST condition, this value is set to NINVerted. Navigation Path: BERT > Test Setup > Unframed BERT > Pattern > TX
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:TX?
<b>Response Syntax</b>	<Pattern>
<b>Response(s)</b>	<b>Pattern:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the polarity type of the transmitter. NINverted, selects Non-Inverted as the polarity. INVerted, selects Inverted as the polarity.
<b>Example(s)</b>	SOUR:DATA:TEL:UPRB:PATT:GLOB:POL:TX INVerted SOUR:DATA:TEL:UPRB:PATT:GLOB:POL:TX? Returns: INVerted
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:TYPE SOURce[1..n]:DATA:TELEcom:PATtern:POLarity

---

**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:RX**

<b>Description</b>	<p>This command sets the global polarity pattern for the receiver.</p> <p>At *RST condition, this value is set to NINVerted.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; RX</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:RX <wsp>NINverted   INVerted
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NINverted   INVerted</p> <p>Selects the polarity for the receiver.</p> <p>NINverted, the polarity selected is Non-Inverted.</p> <p>INVerted, the polarity selected is Inverted.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:UPRB:PATT:GLOB:POL:RX INVerted</p> <p>SENS:DATA:TEL:UPRB:PATT:GLOB:POL:RX?</p> <p>Returns: INVerted</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX?

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:RX?**

<b>Description</b>	<p>This query returns the global polarity pattern for the receiver.</p> <p>At *RST condition, this value is set to NINVerted.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; RX</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:POLarity:RX?
<b>Response Syntax</b>	<Pattern>
<b>Response(s)</b>	<p><b>Pattern:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the polarity type of the transmitter.</p> <p>NINverted, polarity Non-Inverted is selected.</p> <p>INVerted, polarity Inverted is selected.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:UPRB:PATT:GLOB:POL:RX INVerted</p> <p>SENS:DATA:TEL:UPRB:PATT:GLOB:POL:RX?</p> <p>Returns: INVerted</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:PRBS:TYPE:RX

---

---

**:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX**

---

<b>Description</b>	<p>This command sets the test pattern type for the transmitter for the selected lane.</p> <p>At *RST condition, this value is set to PRBs2E23.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; TX</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX &lt;wsp&gt; &lt;Lane&gt;, PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31</p>

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX**

<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted error pattern. The range for the lane is from 0 to 19.</p> <p><b>Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31</p> <p>Sets the test pattern type for the transmitter.</p> <p>PRBs2E9, pattern type is PRBS 2<sup>9</sup>-1.</p> <p>PRBs2E11, pattern type is PRBS 2<sup>11</sup>-1.</p> <p>PRBs2E15, pattern type is PRBS 2<sup>15</sup>-1.</p> <p>PRBs2E20, pattern type is PRBS 2<sup>20</sup>-1.</p> <p>PRBs2E23, pattern type is PRBS 2<sup>23</sup>-1.</p> <p>PRBs2E31, pattern type is PRBS 2<sup>31</sup>-1.</p> <p>PSWAVE1, pattern type is PSWAVE1.</p> <p>PSWAVE2, pattern type is PSWAVE2.</p> <p>PSWAVE4, pattern type is PSWAVE4.</p> <p>PSWAVE8, pattern type is PSWAVE8.</p> <p>PSWAVE16, pattern type is PSWAVE16.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UPRB:PATT:PRBS:TYPE:TX 1, PRBs2E23</p> <p>SOUR:DATA:TEL:UPRB:PATT:PRBS:TYPE:TX? 1</p> <p>Returns: PRBs2E23</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATtern:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue?</p>



**:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX?**

<b>Description</b>	<p>This query returns the test pattern type for the transmitter for the selected lane.</p> <p>At *RST condition, this value is set to PRBs2E23.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; TX</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted error pattern. The range for the lane is from 0 to 19.</p>
<b>Response Syntax</b>	<Pattern>

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SOURce[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:TX?**

<b>Response(s)</b>	<b>Pattern:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the test pattern type for the transmitter for the selected traffic stream. PRBs2E9, pattern type PRBS 2 <sup>9</sup> -1 is selected. PRBs2E11, pattern type PRBS 2 <sup>11</sup> -1 is selected. PRBs2E15, pattern type PRBS 2 <sup>15</sup> -1 is selected. PRBs2E20, pattern type PRBS 2 <sup>20</sup> -1 is selected. PRBs2E23, pattern type PRBS 2 <sup>23</sup> -1 is selected. PRBs2E31, pattern type PRBS 2 <sup>31</sup> -1 is selected. PSWAVE1, pattern type PSWAVE1 is selected. PSWAVE2, pattern type PSWAVE2 is selected. PSWAVE4, pattern type PSWAVE4 is selected. PSWAVE8, pattern type PSWAVE8 is selected. PSWAVE16, pattern type PSWAVE16 is selected.
<b>Example(s)</b>	SOUR:DATA:TEL:UPRB:PATT:PRBS:TYPE:TX 1, PRBs2E23 SOUR:DATA:TEL:UPRB:PATT:PRBS:TYPE:TX? 1 Returns: PRBs2E23
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:TYPE SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:USER:VALue

---

---

**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:RX**

---

**Description**

This command sets the test pattern type for the receiver for the selected lane.

At \*RST condition, this value is set to PRBs2E23.

Navigation Path: BERT > Test Setup > Unframed BERT > Unframed PRBS Pattern Config > RX

**Syntax**

:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:RX <wsp> <Lane>,  
PRBs2E9 | PRBs2E11 | PRBs2E15 | PRBs2E20 | PRBs2E23 | PRBs2E31

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SENSe[1..n]:DATA:TELeom:UPRBs:PATTErn:PRBS:TYPE:RX**

<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted error pattern. The range for the lane is from 0 to 19.</p> <p><b>Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PRBs2E9   PRBs2E11   PRBs2E15   PRBs2E20   PRBs2E23   PRBs2E31</p> <p>Sets the test pattern type for the transmitter.</p> <p>PRBs2E9, pattern type is PRBS 2<sup>9</sup>-1.</p> <p>PRBs2E11, pattern type is PRBS 2<sup>11</sup>-1.</p> <p>PRBs2E15, pattern type is PRBS 2<sup>15</sup>-1.</p> <p>PRBs2E20, pattern type is PRBS 2<sup>20</sup>-1.</p> <p>PRBs2E23, pattern type is PRBS 2<sup>23</sup>-1.</p> <p>PRBs2E31, pattern type is PRBS 2<sup>31</sup>-1.</p> <p>PSWAVE1, pattern type is PSWAVE1.</p> <p>PSWAVE2, pattern type is PSWAVE2.</p> <p>PSWAVE4, pattern type is PSWAVE4.</p> <p>PSWAVE8, pattern type is PSWAVE8.</p> <p>PSWAVE16, pattern type is PSWAVE16.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:UPRB:PATT:PRBS:TYPE:RX 1, PRBs2E23</p> <p>SENS:DATA:TEL:UPRB:PATT:PRBS:TYPE:RX? 1</p> <p>Returns: PRBs2E23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELeom:UPRBs:PATTErn:GLOBal:POLarity:RX?</p>

**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:RX?**

<b>Description</b>	<p>This query returns the test pattern type for the transmitter for the selected lane.</p> <p>At *RST condition, this value is set to PRBs2E23.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; RX</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:PRBS:TYPE:RX? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted error pattern. The range for the lane is from 0 to 19.</p>
<b>Response Syntax</b>	<Pattern>

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SENSe[1..n]:DATA:TELeom:UPRBs:PATtern:PRBS:TYPE:RX?**

<b>Response(s)</b>	<b>Pattern:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the test pattern type for the receiver. PRBs2E9, pattern type PRBS 2 <sup>9</sup> -1 is selected. PRBs2E11, pattern type PRBS 2 <sup>11</sup> -1 is selected. PRBs2E15, pattern type PRBS 2 <sup>15</sup> -1 is selected. PRBs2E20, pattern type PRBS 2 <sup>20</sup> -1 is selected. PRBs2E23, pattern type PRBS 2 <sup>23</sup> -1 is selected. PRBs2E31, pattern type PRBS 2 <sup>31</sup> -1 is selected. PSWAVE1, pattern type PSWAVE1 is selected. PSWAVE2, pattern type PSWAVE2 is selected. PSWAVE4, pattern type PSWAVE4 is selected. PSWAVE8, pattern type PSWAVE8 is selected. PSWAVE16, pattern type PSWAVE16 is selected.
<b>Example(s)</b>	SENS:DATA:TEL:UPRB:PATT:PRBS:TYPE:RX 1, PRBs2E23 SENS:DATA:TEL:UPRB:PATT:PRBS:TYPE:RX? 1 Returns: PRBs2E23
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:UPRBs:PATtern:GLOBal:POLarity:RX

**:SENSe[1..n]:DATA:TELecom:UPRBs:PATtern:GLOBal:COUPled**

<b>Description</b>	<p>This command sets the global pattern for the coupled status for the receiver.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; Coupled Rx to Tx</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:UPRBs:PATtern:GLOBal:COUPled <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the global pattern for coupled status for the receiver.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:UPRB:PATT:GLOB:COUP ON</p> <p>SENS:DATA:TEL:UPRB:PATT:GLOB:COUP?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:UPRBs:PATtern:GLOBal:ALL?

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:COUPled?**

<b>Description</b>	<p>This query returns the global pattern for the coupled status for the receiver.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Unframed BERT &gt; Pattern &gt; Coupled Rx to Tx</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:COUPled?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the coupled status of the receiver.</p> <p>1, Rx to Tx is coupled.</p> <p>0, Rx to Tx is not coupled.</p>
<b>Example(s)</b>	SENS:DATA:TEL:UPRB:PATT:GLOB:COUP?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:GLOBal:ALL

---



---

**:FETCh[1..n]:DATA:TELeom:EOTN:ALARm:LINK?**

---

<b>Description</b>	<p>This command gets the status of EOTN link alarms.</p> <p>For this command, select the Client type as ""Ethernet"" and the interface rate as ""OTU3"" in the Modify Structure setup.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: EOTN Test &gt; Setup &gt; Ether Bert &gt; Link</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:EOTN:ALARm:LINK? <wsp>LOAML1027B   LOBL1027B   HIBER1027B
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOAML1027B   LOBL1027B   HIBER1027B</p> <p>Selects the alarm type whose status is to be retrieved.</p> <p>LOAML1027B  LOBL1027B  HIBER1027B</p>
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the status of Ethernet alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:ALAR:LINK? LOBL1027B
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:PATtern:ALARm:SYNCh?

---

## SCPI Command Reference

*EtherBERT, FC BERT, BERT (CPRI), and Unframed BERT*

---

**:SENSe[1..n]:DATA:TELEcom:SDT:NTTime**

<b>Description</b>	<p>This command selects the no Traffic time (ms) without any Traffic before stopping SDT measurement.</p> <p>At *RST condition, this value is set to 1000.</p> <p>Navigation Path: Test &gt; Ether BERT &gt; Test Configurator &gt; Ether BERT &gt;Service disruption&gt;NTT</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDT:NTTime <wsp> <Time>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Time:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the no traffic time without any traffic before stopping SDT measurement.Choices are from 0.005ms to 1000 ms.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT:NTT 18</p> <p>SENS:DATA:TEL:SDT:NTT?</p> <p>Returns: 18</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDT:SDHSonet:LAYer:TYPE

**:SENSe[1..n]:DATA:TELEcom:SDT:NTTime?**

<b>Description</b>	<p>This query returns the no traffic time (ms) without any traffic before stopping SDT measurement.</p> <p>At *RST condition, this value is set to 10000.</p> <p>Navigation Path: Test &gt; Ether BERT &gt; Test Configurator&gt; Ether BERT &gt;Service disruption&gt;NDT</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDT:NTTime?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional.If no token is specified, the current no traffic time without any traffic is returned.</p>
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the no traffic time without any traffic before stopping SDT measurement.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT:NTT 200</p> <p>SENS:DATA:TEL:SDT:NTT?</p> <p>Returns: 200</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDT:SDHSONet:LAYer:TYPE?

## RFC 2544 - Global

---

:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution

---

<b>Description</b>	<p>This command selects frame size distribution from the list.</p> <p>At *RST condition, this value is set to RFC2544.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution <wsp>RFC2544   UDEFined
<b>Parameter(s)</b>	<p><b>Distribution:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RFC2544   UDEFined</p> <p>Selects frame size distribution from the list.</p> <p>RFC2544: RFC2544</p> <p>UDEFined: User Defined</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FDIS UDEF</p> <p>SOUR:DATA:TEL:ETH:RFC:FDIS?</p> <p>Returns: UDEFined</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution?**

<b>Description</b>	<p>This query returns the frame size distribution from the list.</p> <p>At *RST condition, this value is set to RFC2544.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution?
<b>Response Syntax</b>	<Distribution>
<b>Response(s)</b>	<p><b>Distribution:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the frame size distribution from the list.</p> <p>RFC2544, RFC2544 is returned as frame size distribution.</p> <p>UDEFined, User Defined (UDEFined) is returned as frame size distribution.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FDIS RFC2544</p> <p>SOUR:DATA:TEL:ETH:RFC:FDIS?</p> <p>Returns: RFC2544</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity?</p>

---

## SCPI Command Reference

RFC 2544 - Global

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity**

<b>Description</b>	<p>This command selects quantity of frame size to be used for the test.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity &lt;wsp&gt; &lt;Quantity&gt;</p>
<b>Parameter(s)</b>	<p><b>Quantity:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects quantity of frame size.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:QUAN 7</p> <p>SOUR:DATA:TEL:ETH:RFC:QUAN?</p> <p>Returns: 7</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity?**

---

<b>Description</b>	<p>This query returns the quantity of frame size used for the test.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity?
<b>Response Syntax</b>	<Quantity>
<b>Response(s)</b>	<p><b>Quantity:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the quantity of frame size.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:QUAN 7</p> <p>SOUR:DATA:TEL:ETH:RFC:QUAN?</p> <p>Returns: 7</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIStribution?</p>

---

## SCPI Command Reference

*RFC 2544 - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZe**

<b>Description</b>	<p>This command sets the predefined frame size distribution values for RFC 2544 distribution.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZe &lt;wsp&gt; &lt;Quantity&gt;, &lt;Fsize&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Quantity:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the quantity of frame size.</p> <p><b>Fsize:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the predefined frame size distribution value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FDIS UDEF</p> <p>SOUR:DATA:TEL:ETH:RFC:QUAN 1</p> <p>SOUR:DATA:TEL:ETH:RFC:FSIZ 1, 69</p> <p>SOUR:DATA:TEL:ETH:RFC:FSIZ? 1</p> <p>Returns: 69</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity</p>



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZe?**

<b>Description</b>	<p>This query returns the predefined frame size distribution values for RFC 2544 distribution. At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZe? <wsp> <Quantity>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Quantity:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the quantity of frame size.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current predefined frame size distribution is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Fsize>
<b>Response(s)</b>	<p><b>Fsize:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the predefined frame size distribution values.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FDIS UDEF</p> <p>SOUR:DATA:TEL:ETH:RFC:QUAN 1</p> <p>SOUR:DATA:TEL:ETH:RFC:FSIZ 1, 65</p> <p>SOUR:DATA:TEL:ETH:RFC:FSIZ? 1</p> <p>Returns: 65</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:QUANtity?

---

## SCPI Command Reference

*RFC 2544 - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:RESTore**

<b>Description</b>	<p>This command restores the default configuration for RFC Frame Sizes.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:RESTore
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:REST
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZE

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THRoughput:MINTime?**

---

<b>Description</b>	<p>This query returns the minimum time the throughput subtest is required to run in best condition.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THRoughput:MINTime?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the minimum time the test is required to run in best condition.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:THR:MINT?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:BCKTobck:MINTime?

---

## SCPI Command Reference

*RFC 2544 - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABle**

<b>Description</b>	<p>This command enables or disables the selected RFC 2544 throughput subtest.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABle &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Test:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the throughput test type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABle?**

<b>Description</b>	<p>This query returns the status of the selected throughput subtest.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABle?
<b>Response Syntax</b>	<Get>
<b>Response(s)</b>	<p><b>Get:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the selected subtests.</p> <p>1, is returned when the selected RFC 2544 throughput subtest is enabled.</p> <p>0, is returned when the selected RFC 2544 throughput subtest is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors?

---

## SCPI Command Reference

*RFC 2544 - Global*

---

---

### :FETCh[1..n]:DATA:TELeom:ETHernet:RFC:BCKTobck:MINTime?

---

<b>Description</b>	<p>This query returns the minimum time for the back-to-back subtest required to run in best condition.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:ETHernet:RFC:BCKTobck:MINTime?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the minimum time.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:BCKT:MINT?
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:ETHernet:RFC:THROUGHput:MINTime?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ENABLE**

---

<b>Description</b>	<p>This command enables or disables the selected back-to-back subtest.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ENABLE <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Test:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the back-to-back test type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:BCKT:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:RFC:BCKT:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABLE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABLE?</p>

---

## SCPI Command Reference

RFC 2544 - Global

---

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ENABLE?

---

<b>Description</b>	<p>This query returns the status of the back-to-back subtest.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ENABLE?
<b>Response Syntax</b>	<Get>
<b>Response(s)</b>	<p><b>Get:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the selected RFC 2544 subtests.</p> <p>0, is returned when the status of the selected RFC 2544 subtests is OFF.</p> <p>1, is returned when the status of the selected RFC 2544 subtests is ON.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:BCKT:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:RFC:BCKT:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABLE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABLE?</p>

---



---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:MINTime?**

---

<b>Description</b>	<p>This query returns the minimum time the Frame Loss subtest is required to run in best condition.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:MINTime?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the minimum time.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:FLOS:MINT?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MINTime?

---

## SCPI Command Reference

*RFC 2544 - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:ENABLE**

<b>Description</b>	<p>This command enables or disables the selected Frame Loss subtest.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:ENABLE &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Test:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the frame loss test type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FLOS:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:RFC:FLOS:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABLE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ENABLE?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:ENABLE?**

---

<b>Description</b>	<p>This query returns the status of the Frame Loss subtest.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:ENABLE?
<b>Response Syntax</b>	<Get>
<b>Response(s)</b>	<p><b>Get:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Frame Loss test.</p> <p>0, returns the status of the Frame Loss test as OFF.</p> <p>1, returns the status of the Frame Loss test as ON.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FLOS:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:RFC:FLOS:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TTIME</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TTIME?</p>

---

## SCPI Command Reference

*RFC 2544 - Global*

---

---

### :FETCh[1..n]:DATA:TELeom:ETHernet:RFC:LATency:MINTime?

---

<b>Description</b>	This query returns the minimum time the latency subtest is required to run in best condition. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Setup > Test Configurator > RFC 2544 > Global
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:ETHernet:RFC:LATency:MINTime?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<b>Time:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the minimum time.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:LAT:MINT?
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:ETHernet:RFC:FLOsS:MINimum?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ENABLE**

<b>Description</b>	<p>This command enables or disables the latency subtest.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ENABle <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Test:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the latency subtest.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPY

---

## SCPI Command Reference

*RFC 2544 - Global*

---

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ENABLE?

---

<b>Description</b>	<p>This query returns the status of the latency subtest.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Global</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ENABLE?
<b>Response Syntax</b>	<Get>
<b>Response(s)</b>	<p><b>Get:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the latency subtest.</p> <p>0, returns the latency subtest disabled.</p> <p>1, returns the latency subtest enabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPY?

---

---

**:FETCh[1..n]:DATA:TELeom:ETHernet:RFC:TOTal:MINTime?**

---

<b>Description</b>	This query returns the total time the test is required to run in best condition. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Setup > Test Configurator > RFC 2544 > Global
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:ETHernet:RFC:TOTal:MINTime?
<b>Response Syntax</b>	<Total Time>
<b>Response(s)</b>	<b>Total Time:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the total time the test is required to run in best condition.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:TOT:MINT?
<b>See Also</b>	FETC[1..n]:DATA:TELeom:ETHernet:RFC:LATency:MINimum?

---

## SCPI Command Reference

RFC 2544 - Global

---

---

### :FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:VALidation?

---

<b>Description</b>	This query returns the number of times the result should be validated. At *RST condition, this value is set to 1. Navigation Path: Test > Result > Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:VALidation?
<b>Response Syntax</b>	<Current validation>
<b>Response(s)</b>	<b>Current validation:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of times the result should be validated.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:THR:VAL?
<b>See Also</b>	FETC[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MINTime?

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIRection**

<b>Description</b>	<p>This command sets the Flow Direction.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt;Global Options</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FDIRection &lt;wsp&gt;TXTORX   P1TOP2   FBIDIRCT   LTORemote   RTOLocal</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TXTORX   P1TOP2   FBIDIRCT   LTORemote   RTOLocal</p> <p>Sets the Flow direction</p> <p>TXTORX: TX To RX</p> <p>P1TOP2: Port #1 To Port #2</p> <p>FBIDIRCT: Bidirectional</p> <p>LTORemote: Local To Remote</p> <p>RTOLocal: Remote To Local</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:RFC:FDIRection TXTORX</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:FCOunt:TX</p>

---

## SCPI Command Reference

*RFC 2544 - Global*

---

---

### :SOURce[1..n]:DATA:TELecom:ETHernet:RFC:FDIRection?

---

<b>Description</b>	This query returns the Flow Direction. At *RST condition, this value is set to 1. Navigation Path: Test > Setup > Test Configurator > RFC 2544 > Global Options
<b>Syntax</b>	:SOURce[1..n]:DATA:TELecom:ETHernet:RFC:FDIRection?
<b>Response Syntax</b>	<direction>
<b>Response(s)</b>	<b>direction:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Flow direction
<b>Example(s)</b>	SOURce:DATA:TELecom:ETHernet:RFC:FDIRection?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:LATency:FCOunt:TX?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:DUALtest:ENABled**

<b>Description</b>	<p>This command sets Dual Test Set status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Dual Test Set</p> <p>Navigation Path: Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Dual Test Set</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:DUALtest:ENABled <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the sets Dual Test Set status.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:DUAL:ENAB ON
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:CONNect?

---

## SCPI Command Reference

*RFC 2544 - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:DUALtest:ENABled?**

<b>Description</b>	<p>This query returns the Dual Test Set status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Dual Test Set</p> <p>Navigation Path: Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Dual Test Set</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:DUALtest:ENABled?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Dual Test Set status.</p> <p>1, Dual Test Set status is enabled.</p> <p>0, Dual Test Set status is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:DUAL:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:DUAL:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:REMOte:CONNect

# Smart Loopback

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MODE**

---

<b>Description</b>	<p>This command selects the mode for Smart Loopback application.</p> <p>At *RST condition, this value is set to UDP.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Smart Loopback &gt; Loopback &gt; Mode</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MODE &lt;wsp&gt;UDPTCP   IP   ETHERNET   EAUNICAST (All Unicast)</code>
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: UDPTCP   IP   ETHERNET   EAUNICAST (All Unicast)</p> <p>Selects the loopback mode for the smart loopback application.</p> <p>UDPTCP, UDP mode is selected.</p> <p>IP, IP mode is selected.</p> <p>ETHERNET, Ethernet mode is selected.</p> <p>EAUNICAST, Ethernet all unicast mode is selected.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:SLO:MODE IP</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:NETWork:VLAN:ID</code>

---

## SCPI Command Reference

### Smart Loopback

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MODE?**

<b>Description</b>	<p>This query returns the selected mode for Smart Loopback application.</p> <p>At *RST condition, this value is set to UDP.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Smart Loopback &gt; Loopback &gt; Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the loopback mode.</p> <p>IP, IP mode is selected.</p> <p>UDPTCP, UDP mode is selected.</p> <p>ETHERNET, Ethernet mode is selected.</p> <p>EAUNICAST, Ethernet all unicast mode is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:SLO:MODE IP</p> <p>SOUR:DATA:TEL:ETH:SLO:MODE?</p> <p>Returns: IP</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:NETWork:VLAN:ID?

**:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MATCHing:UDP:PORT:MODE?**

<b>Description</b>	<p>This query returns the UDP port value of Smart Loopback application.</p> <p>At *RST condition, this value is set to All.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Smart Loopback &gt; Loopback &gt; Matching &amp; Swapping &gt; UDP Port</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MATCHing:UDP:PORT:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the UDP port mode.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:SLO:MATC:UDP:PORT:MODE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MODE?

---

## SCPI Command Reference

### Smart Loopback

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MATChing:MAC:ADDRess:MODE?**

<b>Description</b>	<p>This query returns the UDP port value of Smart Loopback application.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Smart Loopback &gt; Loopback &gt; Matching &amp; Swapping &gt; MAC Address</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MATChing:MAC:ADDRess:MODE?</code>
<b>Response Syntax</b>	<code>&lt;Mode&gt;</code>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the MAC address.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:SLO:MATC:MAC:ADDR:MODE?</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:MODE?</code>

---



# BERT

---

**:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:SYNC?**

---

<b>Description</b>	<p>This query returns the Synch status of pattern alarm for the selected lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: BERT &gt; Test &gt; Setup &gt; Unframed BERT &gt; Sync</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:SYNC? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane of pattern alarm.</p>
<b>Response Syntax</b>	<Sync>
<b>Response(s)</b>	<p><b>Sync:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Synch status of the pattern alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PATT:ALAR:SYNC? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:GLOBal:ALARm:SYNC?

---

## SCPI Command Reference

### BERT

---

---

#### :FETCh[1..n]:DATA:TELeom:PATTer:n:GLOBAl:ALARm:SYNC?

---

<b>Description</b>	<p>This query returns the Globally Synch status of the pattern alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: BERT &gt; Test &gt; Setup &gt; Unframed BERT &gt; Global Sync</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:PATTer:n:GLOBAl:ALARm:SYNC?
<b>Response Syntax</b>	<Sync>
<b>Response(s)</b>	<p><b>Sync:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Globally Synch status of the pattern alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PATT:GLOB:ALAR:SYNC?
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:PATTer:n:ALARm:SYNC?

---

---

**:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:RATE**

---

<b>Description</b>	<p>This command sets the BERT Threshold rate value.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; BERT &gt;Pattern &gt;BERT Threshold</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:RATE &lt;wsp&gt; &lt;Set&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Pattern Threshold Rate,</p> <p>Choices are 1.0E-14 through 1.9E-01</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:PATtern:THReshold:RATE 1.0E-12</pre> <pre>SENS:DATA:TEL:PATtern:THReshold:RATE?</pre> <p>Returns: 1.0E-12</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:SDT:PATtern:THReshold:COUNT?</pre>

---

## SCPI Command Reference

### BERT

---

**:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:RATE?**

<b>Description</b>	<p>This query returns the the BERT Threshold rate value.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; BERT &gt;Pattern &gt;BERT Threshold</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:RATE?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional.</p> <p>If no token is specified, thePattern Threshold Rate is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the pattern BER Threshold rate.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:PATtern:THReshold:RATE 1.0E-12</p> <p>SENS:DATA:TEL:PATtern:THReshold:RATE?</p> <p>Returns: 1.0E-12</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:SDT:PATtern:THReshold:COUNT</p>

---

**:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:COUNT**

---

<b>Description</b>	<p>This command sets the BERT Threshold count value.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; BERT &gt; Pattern &gt; BERT Threshold</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:COUNT &lt;wsp&gt; &lt;Set&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Pattern Threshold Count Value</p> <p>Choices are 0 to 999999</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:PATtern:THReshold:COUN 1000</p> <p>SENS:DATA:TEL:PATtern:THReshold:COUN?</p> <p>Returns: 1000</p>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:SDT:NDTime?</code>

---

## SCPI Command Reference

### BERT

---

#### :SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:COUNT?

<b>Description</b>	<p>This query returns the BERT Threshold count value.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; BERT &gt; Pattern &gt; BERT Threshold</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:PATtern:THReshold:COUNT?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional.</p> <p>If no token is specified, the Pattern Threshold Count Value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the pattern Threshold Count.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:PATtern:THReshold:COUN 1000</p> <p>SENS:DATA:TEL:PATtern:THReshold:COUN?</p> <p>Returns: 1000</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:SDT:NDTime</p>

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:VERDict:DISable**

---

<b>Description</b>	<p>This command resets status of BERT Threshold.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path:Test Setup &gt; Test Configurator &gt; BERT &gt; Pattern &gt;Pass/Fail Verdict&gt;Disabled</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:VERDict:DISable
<b>Example(s)</b>	SOUR:DATA:TEL:PATtern:VERDict:DISable
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:DEFault?

---

## SCPI Command Reference

### BERT

---

---

#### :SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:RATE

---

<b>Description</b>	<p>This command sets the BERT Threshold rate value.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Unframed BERT &gt;Pattern &gt;BERT Threshold</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:RATE &lt;wsp&gt; &lt;Set&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Pattern Threshold Rate,</p> <p>Choices are 1.0E-14 through 1.9E-01</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:UPRB:PATtern:THReshold:RATE 1.0E-12 SENS:DATA:TEL:UPRB:PATtern:THReshold:RATE? Returns: 1.0E-12</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:SDT:PATtern:THReshold:COUNt?</pre>

---



**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:RATE?**

<b>Description</b>	<p>This query returns the BERT Threshold rate value.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Unframed BERT &gt;Pattern &gt;BERT Threshold</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:RATE?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional.</p> <p>If no token is specified, the Pattern Threshold Rate is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the pattern Threshold rate.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:UPRB:PATtern:THReshold:RATE 1.0E-12</p> <p>SENS:DATA:TEL:UPRB:PATtern:THReshold:RATE?</p> <p>Returns: 1.0E-12</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDT:PATtern:THReshold:COUNT

## SCPI Command Reference

### BERT

---

**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:COUNT**

<b>Description</b>	<p>This command sets the BERT Threshold count value.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Unframed BERT &gt; Pattern &gt; BERT Threshold</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:COUNT &lt;wsp&gt; &lt;Set&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Pattern Threshold Count Value</p> <p>Choices are 0 to 999999</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:UPRB:PATtern:THReshold:COUN 1000</p> <p>SENS:DATA:TEL:UPRB:PATtern:THReshold:COUN?</p> <p>Returns: 1000</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:SDT:NDTime?</p>

**:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:COUNt?**

<b>Description</b>	<p>This query returns the BERT Threshold count value.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Unframed BERT &gt; Pattern &gt; BERT Threshold</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:COUNt?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional.</p> <p>If no token is specified, the Pattern Threshold Count Value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the pattern Threshold Count.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:UPRB:PATtern:THReshold:COUN 1000</p> <p>SENS:DATA:TEL:UPRB:PATtern:THReshold:COUN?</p> <p>Returns: 1000</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDT:NDTime

## SCPI Command Reference

### *BERT*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:REStore:DEFault**

<b>Description</b>	This command resets or overwrites the Optical Payload Unit (OPU) overhead byte values. This command is an event and is not associated with an *RST condition or a query form. Navigation Path: OTN Test > Setup > Bert > Restore OTN Bert Defaults
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:REStore:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:REST:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:RESet

---

---

**:SENSe[1..n]:DATA:TELEcom:SDT:NDTime**

---

<b>Description</b>	<p>This command selects the no defect time (ms) without any defects before stopping SDT measurement.</p> <p>At *RST condition, this value is set to 300 ms.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Test Configurator &gt; Bert &gt;Service disruption&gt;NDT</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDT:NDTime <wsp> <Time>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Time:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element. The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the no defect time without any defects before stopping SDT measurement.Choices are from 0.005ms to 2000 ms.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT:NDT 18</p> <p>SENS:DATA:TEL:SDT:NDT?</p> <p>Returns: 18</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE

---

## SCPI Command Reference

### BERT

---

#### :SENSe[1..n]:DATA:TELEcom:SDT:NDTime?

<b>Description</b>	<p>This query returns the no defect time (ms) without any defects before stopping SDT measurement.</p> <p>At *RST condition, this value is set to 300 ms.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Test Configurator &gt; Bert &gt;Service disruption&gt;NDT</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDT:NDTime?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current no defect time without any defects is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the no defect time without any defects before stopping SDT measurement.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT:NDT 200</p> <p>SENS:DATA:TEL:SDT:NDT?</p> <p>Returns: 200</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE?

---

**:SENSe[1..n]:DATA:TELEcom:SDT**

<b>Description</b>	<p>This command enables or disables the disruption time measurements.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Test Configurator&gt;Bert &gt;Service disruption &gt; disruption Monitoring</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDT <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the disruption time measurements.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT ON</p> <p>SENS:DATA:TEL:SDT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection</p>

---

## SCPI Command Reference

### BERT

---

**:SENSe[1..n]:DATA:TELEcom:SDT?**

<b>Description</b>	<p>This query returns the status of disruption time measurements.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Test Configurator&gt;Bert &gt;Service disruption &gt; disruption Monitoring</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDT?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the disruption time measurements.</p> <p>1 - measurement is enabled.</p> <p>0 - measurement is disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT ON</p> <p>SENS:DATA:TEL:SDT?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE?

---



**:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE**

<b>Description</b>	<p>This command selects on which layer the service disruption time test is performed for OTN. At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Test Configurator&gt;Bert &gt;Service disruption &gt;Defect</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE <wsp>PATtern   OTL   OTU3e1   ODU3e1   OPU3e1   FEC   INTerface   OTU3e2   ODU3e2   OPU3e2   OTU3   ODU3   OPU3   OTU4   ODU4   OPU4   SECTION   LINE   HOP   ODU2   OPU2   ODU1   OPU1   ODU0   OPU0
<b>Parameter(s)</b>	<p><b>Layer:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PATtern   OTL   OTU3e1   ODU3e1   OPU3e1   FEC   INTerface   OTU3e2   ODU3e2   OPU3e2   OTU3   ODU3   OPU3   OTU4   ODU4   OPU4   SECTION   LINE   HOP   ODU2   OPU2   ODU1   OPU1   ODU0   OPU0</p> <p>Selects on which layer the service disruption time test is performed. Choices depend on the selected test path.</p> <p>FEC, OPU3E1, ODU3E1, OTU3E1, PATtern, OTL, INTerface, OTU3e2, ODU3e2, OPU3e2, OTU3, ODU3, OPU3, OTU4, ODU4, OPU4, SECTION, LINE, or HOP.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT:OTN:LAY:TYPE OTU3e1</p> <p>SENS:DATA:TEL:SDT:OTN:LAY:TYPE</p> <p>Returns: OTU3e1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection

## SCPI Command Reference

### *BERT*

---

**:SENSe[1..n]:DATA:TELeCom:SDT:OTN:LAYer:TYPE?**

<b>Description</b>	<p>This query returns the layer for service disruption time test for OTN.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Test Configurator&gt;Bert &gt;Service disruption &gt;Defect</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeCom:SDT:OTN:LAYer:TYPE?
<b>Response Syntax</b>	<Layer>
<b>Response(s)</b>	<p><b>Layer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the layer for service disruption time test is performed.</p> <p>FEC, OPU3E1, ODU3E1, OTU3E1, PATtern, OTL, INTerface, OTU3e2, ODU3e2, OPU3e2, OTU3, ODU3, OPU3, OTU4, ODU4, OPU4, SECTION, LINE, or HOP.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT:OTN:LAY:TYPE Interface</p> <p>SENS:DATA:TEL:SDT:OTN:LAY:TYPE?</p> <p>Returns: Interface</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELeCom:SDT:OTN:DSELection?

---

**:SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection**

<b>Description</b>	<p>This command selects the defect of OTN for specific layer.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Test Configurator&gt;Bert &gt;Service disruption &gt;Defect</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection &lt;wsp&gt;PLOS   BITERROR   OPU AIS   OPUCSF   OTLLOF   OTLOOF   OTLLOL   OTLLOR   OTLOOR   INVMARKER   OTLFAS   FCCW   FUCW   OAIS   OBDI   OLOF   OOF   OLOM   OOM   OBIAE   OIAE   OBIP8   OBEI   FAS   MFAS   ODAIS   ODBDI   OLCK   OOCI   OFSF   OBSF   OFSD   OBSD   ODBIP8   OD BEI   LosO   OMSIM   OPULOOMFI   OPUOOMFI   OPUOMFI   OPUPLM   LOF   B1   AISL   MSRDI   MSR   B2   AUA   HPRD   HPR   PDIP   B3   HPUNEQ   LOFL</pre>
<b>Parameter(s)</b>	<p><b>Selection:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PLOS   BITERROR   OPUAIS   OPUCSF   OTLLOF   OTLOOF   OTLLOL   OTLLOR   OTLOOR   INVMARKER   OTLFAS   FCCW   FUCW   OAIS   OBDI   OLOF   OOF   OLOM   OOM   OBIAE   OIAE   OBIP8   OBEI   FAS   MFAS   ODAIS   ODBDI   OLCK   OOCI   OFSF   OBSF   OFSD   OBSD   ODBIP8   OD BEI   LosO   OMSIM   OPULOOMFI   OPUOOMFI   OPUOMFI   OPUPLM   LOF   B1   AISL   MSRDI   MSR   B2   AUA   HPRD   HPR   PDIP   B3   HPUNEQ   LOFL</p> <p>Selects the defect for OTN layer.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:SDT:OTN:DSEL BITERROR SENS:DATA:TEL:SDT:OTN:DSEL?</pre> <p>Returns: BITERROR</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE</pre>

## SCPI Command Reference

### BERT

---

#### :SENSe[1..n]:DATA:TELeom:SDT:OTN:DSElection?

<b>Description</b>	<p>This query returns the defect for specific layer for OTN.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Test Configurator&gt;Bert &gt;Service disruption &gt;Defect</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:SDT:OTN:DSElection?
<b>Response Syntax</b>	<Selection>
<b>Response(s)</b>	<p><b>Selection:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the defect for OTN layer.</p> <p>PLOS, BITERROR, OPUAIS, OPUCSF, OTLLOF, OTLOOF, OTLLOL, OTLLOR, OTLOOR, INVMARKER, OTLFAS, FCCW, FUCW, OAIS, OBDI, OLOF, OOOF, OLOM, OOM, OBIAE, OIAE, OBIP8, OBEI, FAS, MFAS, ODAIS, ODBDI, OLCK, OOCI, OFSF, OBSF, OFSD, OBSD, ODBIP8, ODBEI, LosO, OMSIM, OPULOOMFI, OPUOOMFI, OPUOMFI, OPUPLM, RLOF, RFAS, B1, SEF, AISL, REIL, RDIL, B2, AISP, AULOP, EPCD, EPPD, EPSD, RDIP, REIP, PDIP, B3, HPUNEQ, PLM, HPREI, AISV, LPRDI, EVSD, EVCD, EVPD, LPRFI, TULOP, LOPTIM, LOPPLM, LOPUNEQ, BIP2, LPREI, TU3AIS, TU3RDI, TU3PSD, TU3PCD, TU3PPD, TU3LOM, TU3LOP, TU3TIM, TU3PLM, TU3PDI, TU3UNEQ, TU3B3, TU3REI, TU3UNEQ, HOPTCMLTC, HOPTCMUNEQ, HOPTCMIAIS, HOPTCMRDI, HOPTCMODI, HOPTCMIEC, HOPTCMREI, HOPTCMOEI, HOPTCMVIOL, LOPTCMLTC, LOPTCMIAIS, LOPTCMRDI, LOPTCMODI, LOPTCMREI, LOPTCMOEI, LOPTCMVIOL, LOPTCMUNEQ, TU3TCMLTC, TU3TCMIAIS, TU3TCMRDI, TU3TCMODI, TU3TCMIEC, TU3TCMREI, TU3TCMOEI, TU3TCMVIOL, TU3TCMUNEQ.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT:OTN:DSEL BITERROR</p> <p>SENS:DATA:TEL:SDT:OTN:DSEL?</p> <p>Returns: BITERROR</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:SDT:OTN:LAYer:TYPE?

---

**:SENSe[1..n]:DATA:TELEcom:SDT:VERDict**

---

<b>Description</b>	<p>This command sets the Pass/Fail Verdict for the instrument.</p> <p>At *RST condition, this value is set to Enabled.</p> <p>Navigation Path: Test &gt; Ether BERT &gt; Test Configurator &gt; Ether BERT &gt;Service disruption&gt;PassFail Verdict&gt;PassFail Verdict</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDT:VERDict <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Pass/Failed Verdict.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT:VERD ON</p> <p>SENS:DATA:TEL:SDT:VERD?</p> <p>Returns: 0</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:SDT:SDHSonet:LAYer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:SDT:SDHSonet:DSELection</p>

---

## SCPI Command Reference

### BERT

---

---

#### :SENSe[1..n]:DATA:TELecom:SDT:VERDict?

---

<b>Description</b>	<p>This query returns the Pass/Fail Verdict for the instrument.</p> <p>At *RST condition, this value is set to Enabled.</p> <p>Navigation Path: Test &gt; Ether BERT &gt; Test Configurator &gt; Ether BERT &gt;Service disruption&gt;PassFail Verdict&gt; PassFail Verdict</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:SDT:VERDict?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Pass/Failed Verdict status</p> <p>1 - Pass/Failed Verdict status is enabled.</p> <p>0 - Pass/Failed Verdict status is disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT:VERD ON</p> <p>SENS:DATA:TEL:SDT:VERD?</p> <p>Returns: 0</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:SDT:SDHSonet:LAYer:TYPE?

---

---

**:SENSe[1..n]:DATA:TELEcom:SDT:THReshold**

---

<b>Description</b>	<p>This command selects the configurable threshold value to declare the Service Disruption Pass/Fail verdict.</p> <p>At *RST condition, this value is set to 50 ms.</p> <p>Navigation Path: Test &gt; BERT &gt; Test Configurator &gt; Ether BERT &gt;Service disruption&gt;threshold</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:SDT:THReshold &lt;wsp&gt; &lt;Threshold&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Threshold:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the configurable threshold value to declare the Service Disruption Pass/Fail verdict.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:SDT:THR 18 SENS:DATA:TEL:SDT:THR? Returns: 18</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI</code>

---

## SCPI Command Reference

### BERT

---

#### :SENSe[1..n]:DATA:TELEcom:SDT:THReshold?

<b>Description</b>	<p>This query returns the configurable threshold value to declare the Service Disruption Pass/Fail verdict.</p> <p>At *RST condition, this value is set to 50 ms.</p> <p>Navigation Path: Test &gt; BERT &gt; Test Configurator &gt; Ether BERT &gt;Service disruption&gt;threshold</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDT:THReshold?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>The configurable threshold value is returned.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the configurable threshold value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDT:THR 18</p> <p>SENS:DATA:TEL:SDT:THR?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI?

---



## Interface - Laser ON/OFF

---

:SENSe[1..n]:DATA:TELEcom:ALASer

---

<b>Description</b>	<p>This command enables or disables the state of the all lanes laser.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; Laser ON/OFF &gt; All Lanes</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ALASer <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the laser for all Lanes.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ALAS ON</p> <p>SENS:DATA:TEL:ALAS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ALASer ON</p> <p>SENSe[1..n]:DATA:TELEcom:ALASer?</p> <p>Returns: 1</p>

---

## SCPI Command Reference

### Interface - Laser ON/OFF

---

:SENSe[1..n]:DATA:TELEcom:ALASer?

<b>Description</b>	<p>This query returns the current status of all lanes laser.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; Laser ON/OFF &gt; All Lanes</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ALASer?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the laser for all lanes.</p> <p>1, returns the status of the laser for all lanes as ON.</p> <p>0, returns the status of the laser for all lanes as OFF.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ALAS ON</p> <p>SENS:DATA:TEL:ALAS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:LASer?</p> <p>SENSe[1..n]:DATA:TELEcom:LASer</p>

---

**:SENSe[1..n]:DATA:TELEcom:LASer**

<b>Description</b>	<p>This command enables or disables the state of the laser for the selected traffic stream.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; Laser ON/OFF &gt; Per Lane</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:LASer &lt;wsp&gt; &lt;Lane&gt;, ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the current status of an Optical lane. The range for the Optical lane is from 0 to 9.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the laser according to per lane configuration.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:LAS 1, ON SENS:DATA:TEL:LAS? 1 Returns: 1</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:LASer? SENSe[1..n]:DATA:TELEcom:ALASer</pre>

---

## SCPI Command Reference

### Interface - Laser ON/OFF

---

:SENSe[1..n]:DATA:TELEcom:LASer?

<b>Description</b>	<p>This query returns the current status of laser for the selected traffic stream.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface &gt; Physical Interface &gt; Laser ON/OFF &gt; Per Lane</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:LASer? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the current status of an Optical lane. The range for the Optical lane is from 0 to 9.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the laser for the selected traffic stream.</p> <p>1, returns the status of the laser for the selected traffic stream as ON.</p> <p>0, returns the status of the laser for the selected traffic stream as OFF.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:LAS 1, ON</p> <p>SENS:DATA:TEL:LAS? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ALASer

---

## Signal - Signal Configuration (OTN) - Modify Tributary Slots/Port

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABLE**

---

<b>Description</b>	<p>This command sets the fixed structure status.</p> <p>For this command, in the "Modify Structure" block of the GUI, in the "OTN Multiplexing" dropdown select values other than "ODU3" and "ODU4" so that the "Modify Tributary Slots" button is enabled.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4/10 lanes) &gt; Modify Tributary Slots &gt; Fixed Structure</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABle <wsp> <OduType>, ON   OFF
<b>Parameter(s)</b>	<p><b>OduType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the ODU type value.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the fixed structure.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:FSTR:ENAB "ODU3",ON
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:PORT

---

## SCPI Command Reference

### *Signal - Signal Configuration (OTN) - Modify Tributary Slots/Port*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABLE?**

<b>Description</b>	<p>This query returns the fixed structure status.</p> <p>For this command, in the ""Modify Structure"" block of the GUI, in the ""OTN Multiplexing"" dropdown select values other than ""ODU3"" and ""ODU4"" so that the ""Modify Tributary Slots"" button is enabled.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU(4/10 lanes) &gt; Modify Tributary Slots &gt; Fixed Structure</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABLE? <wsp> <OduType>
<b>Parameter(s)</b>	<p><b>OduType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the ODU type value.</p>
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<p><b>set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the fixed structure status.</p> <p>1, fixed structure status is enabled.</p> <p>0, fixed structure status is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:FSTR:ENAB? "ODU3"
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:PORT

---

**:SOURce[1..n]:DATA:TELEcom:OTN:PORT**

<b>Description</b>	<p>This command sets the port value.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4/10 lanes) &gt; Modify Tributary Slots &gt; Tributary Port</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:PORT <wsp> <OduType>, <Set>
<b>Parameter(s)</b>	<p><b>OduType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the ODU type value.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the port value.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:PORT "ODU3",3
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABle

## SCPI Command Reference

### *Signal - Signal Configuration (OTN) - Modify Tributary Slots/Port*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:PORT?**

<b>Description</b>	<p>This query returns the port value.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4/10 lanes) &gt; Modify Tributary Slots &gt; Fixed Structure</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:PORT? &lt;wsp&gt; &lt;OduType&gt;</code>
<b>Parameter(s)</b>	<p><b>OduType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the ODU type value.</p>
<b>Response Syntax</b>	<code>&lt;port&gt;</code>
<b>Response(s)</b>	<p><b>port:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the port value.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:PORT? "ODU3"</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABle</code>

---



:SOURCE[1..n]:DATA:TELEcom:OTN:POSITION

<b>Description</b>	<p>This command sets the slot value.</p> <p>For this command, in the ""Modify Structure"" block of the GUI, in the ""OTN Multiplexing"" dropdown select values other than ""ODU3"" and ""ODU4"" so that the ""Modify Tributary Slots"" button is enabled.</p> <p>At *RST condition, this value is set to 1(ON).</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4/10 lanes) &gt; Modify Tributary Slots</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:POSITION <wsp> <OduType>, <Slot>, ON   OFF
<b>Parameter(s)</b>	<p><b>OduType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the ODU type value.</p> <p><b>Slot:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the slot value.</p> <p><b>Set:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the slot status whether selected or not selected.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:POS "ODU2",1, ON
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABLE

## SCPI Command Reference

### Signal - Signal Configuration (OTN) - Modify Tributary Slots/Port

---

**:SOURce[1..n]:DATA:TELEcom:OTN:POSition?**

<b>Description</b>	<p>This query returns the slot value.</p> <p>For this command, in the ""Modify Structure"" block of the GUI, in the ""OTN Multiplexing"" dropdown select values other than ""ODU3"" and ""ODU4"" so that the ""Modify Tributary Slots"" button is enabled.</p> <p>At *RST condition, this value is set to 1(ON).</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4/10 lanes) &gt; Modify Tributary Slots</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:POSition? <wsp> <OduType>, <Slot>
<b>Parameter(s)</b>	<p><b>OduType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the ODU type value.</p> <p><b>Slot:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the slot value.</p>
<b>Response Syntax</b>	<status>
<b>Response(s)</b>	<p><b>status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the slot status, whether it is selected or not.</p> <p>1, returns the slot value as ON.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:POS? "ODU2",1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABLE

---

**:SOURce[1..n]:DATA:TELEcom:OTN:POSition:RANGe**

---

<b>Description</b>	<p>This command sets the slot position status for specified ODU type for given range.</p> <p>Command is for configuring multiple positions at a time and works only for configurations where Clear all and default buttons are available.</p> <p>At *RST condition, this value is set to 1 (ON).</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4/10 lanes) with multiplexing &gt; Modify Tributary Slots</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:POSition:RANGe <wsp> <OduType>, <Start slot>, <End slot>, ON   OFF
<b>Parameter(s)</b>	<p><b>OduType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the ODU type for which the positions are to be modified.</p> <p><b>Start slot:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the start position</p> <p><b>End slot:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the end position</p> <p><b>Status:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:POS:RANG ODU4,1,31,1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:POSition

---

## SCPI Command Reference

### *Signal - Signal Configuration (OTN) - Modify Tributary Slots/Port*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:POSition:RANGe?**

<b>Description</b>	<p>This query returns the slot status for all positions, for selected ODU types.</p> <p>At *RST condition, this value is set to 1 (ON).</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4/10 lanes) &gt; Modify Tributary Slots</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:POSition:RANGe? &lt;wsp&gt; &lt;OduType&gt;</code>
<b>Parameter(s)</b>	<p><b>OduType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p>
<b>Response Syntax</b>	<code>&lt;Slot status&gt;</code>
<b>Response(s)</b>	<p><b>Slot status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns all available slots and their status for specified ODU type.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:POS:RANG ODU4,1,31,1</code> <code>SOUR:DATA:TEL:OTN:POS:RANG? ODU4</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:POSition?</code>

---

---

**:FETCh[1..n]:DATA:TELecom:OTN:BITRate?**

---

<b>Description</b>	<p>This query returns the nominal bit rate value.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4 lanes) &gt; Modify Tributary Slots</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:BITRate?
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the nominal bit rate value</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:BITRate?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:SLOTs

---

## SCPI Command Reference

### *Signal - Signal Configuration (OTN) - Modify Tributary Slots/Port*

---

**:FETCh[1..n]:DATA:TELEcom:OTN:SLOTS?**

<b>Description</b>	This query returns the nominal bit rate value. At *RST condition, this value is set to 1. Navigation Path: OTN BERT Test > Test > Setup > OTU (4 lanes) > Modify Tributary Slots
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:SLOTS?
<b>Response Syntax</b>	<Slots>
<b>Response(s)</b>	<b>Slots:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Count of No.of Tributary Slots selected.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:SLOTS?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:BITRate

---

## Signal - Signal Configuration (OTN) - Config TCM

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:CONFig:TCM[1..n]**

---

<b>Description</b>	<p>This command enables or disables the TCM values.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU &gt; Signal Configuration &gt; Config TCM</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:CONFig:TCM[1..n] &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the TCM values.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:CONF:TCM1 ON</p> <p>SOUR:DATA:TEL:OTN:ODU3:CONF:TCM1?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler?</p>

---

## SCPI Command Reference

### *Signal - Signal Configuration (OTN) - Config TCM*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:CONFig:TCM[1..n]?**

<b>Description</b>	<p>This query returns the status of TCM values.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU &gt; Signal Configuration &gt; Config TCM</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:CONFig:TCM[1..n]?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the TCM values.</p> <p>0, returns the status as OFF.</p> <p>1, returns the status as ON.</p>
<b>Example(s)</b>	<p>OUR:DATA:TEL:OTN:ODU100:CONF:TCM1 ON</p> <p>SOUR:DATA:TEL:OTN:ODU3:CONF:TCM1?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler?</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:CONFig:TCM[1..n]**

---

<b>Description</b>	<p>This command sets the status of TCM values.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU3e(1/2) &gt; Signal Configuration &gt; Config TCM</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; ODU(1/2)e &gt; Signal Configuration &gt; Config TCM</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:CONFig:TCM[1..n] <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the TCM values.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:CONF:TCM1 ON</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:CONF:TCM1?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler?</p>

---

## SCPI Command Reference

### *Signal - Signal Configuration (OTN) - Config TCM*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:CONFig:TCM[1..n]?**

<b>Description</b>	<p>This query returns the status of TCM values.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; OTU3e(1/2) &gt; Signal Configuration &gt; Config TCM</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; ODU(1/2)e &gt; Signal Configuration &gt; Config TCM</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:CONFig:TCM[1..n]?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the TCM values.</p> <p>0, returns the status as OFF.</p> <p>1, returns the status as ON.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ODU3:E1:CONF:TCM1 OFF</p> <p>SOUR:DATA:TEL:OTN:ODU3:E1:CONF:TCM1?</p> <p>Returns: 0</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:SCRambler?</p>

---

# Modify Frame Structure

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DATAlink**

---

<b>Description</b>	<p>This command sets the data link type for the selected traffic stream.</p> <p>At *RST condition, this value is set to ETHernet.</p> <p>Navigation Path: BERT &gt; TestSetup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Modify frame structure &gt; Frame format</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DATAlink &lt;wsp&gt; &lt;Tgen&gt;, ETHERNETII</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Datalink:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: ETHERNETII</p> <p>Set the type of datalink.</p> <p>ETHernet</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DAT 1, ETH</p> <p>SOUR:DATA:TEL:ETH:STR:DAT? 1</p> <p>Returns: ETHERNET</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DATAlink?</p>

---

## SCPI Command Reference

### *Modify Frame Structure*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DATalink?**

<b>Description</b>	<p>This query returns the value of data link type for the selected traffic stream.</p> <p>At *RST condition, this value is set to ETHERNETII.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Modify Frame Structure &gt; Frame format</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DATalink? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Datalink>
<b>Response(s)</b>	<p><b>Datalink:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the selected datalink type.</p> <p>ETHERNETII, ETHERNETII as datalink is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DAT 1, ETH</p> <p>SOUR:DATA:TEL:ETH:STR:DAT? 1</p> <p>Returns: ETHERNETII</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DATalink</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork**

---

<b>Description</b>	<p>This command sets the network type for the selected traffic stream.</p> <p>At *RST condition, this value is set to IPV4.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Modify Frame Structure &gt; Network Layer</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork &lt;wsp&gt; &lt;Tgen&gt;, NONE   IPV4   IPV6   MPLSIPV4   MPLSIPV6</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Network:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NONE   IPV4   IPV6   MPLSIPV4   MPLSIPV6</p> <p>Selects the type of network.</p> <p>NONE: None</p> <p>IPV4: Internet Protocol version 4</p> <p>IPV6: Internet Protocol version 6</p> <p>MPLSIPV4: IPV4 (Internet Protocol version 4) when MPLS is enabled</p> <p>MPLSIPV6: IPV6 (Internet Protocol version 6) when MPLS is enabled</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:NETW 1, IPV4</p> <p>SOUR:DATA:TEL:ETH:STR:NETW?</p> <p>Returns: IPV4</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DATAlink</p>

---

## SCPI Command Reference

### *Modify Frame Structure*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork?**

<b>Description</b>	<p>This query returns the network traffic type for the selected traffic stream.</p> <p>At *RST condition, this value is set to IPV4.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Modify Frame Structure &gt; Network Layer</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Network>
<b>Response(s)</b>	<p><b>Network:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the selected network type.</p> <p>NONE, No network type is selected.</p> <p>IPV4, IPV4 is selected as the network type.</p> <p>IPV6, IPV6 is selected as the network type.</p> <p>MPLSIPV4, MPLSIPV4 is selected as the network type.</p> <p>MPLSIPV6,MPLSIPV6 is selected as the network type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:NETW 1, IPV4</p> <p>SOUR:DATA:TEL:ETH:STR:NETW? 1</p> <p>Returns: IPV4</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsport?**

<b>Description</b>	<p>This query returns the transport type for the selected traffic stream.</p> <p>At *RST condition, this value is set to UDP.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Modify Frame Structure &gt; Transport Layer</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsport? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Transport>
<b>Response(s)</b>	<p><b>Transport:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the selected transport type.</p> <p>UDP, transport link as UDP is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TRAN 1, UDP</p> <p>SOUR:DATA:TEL:ETH:STR:TRAN? 1</p> <p>Returns: UDP</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:NETWork

---

## SCPI Command Reference

### *Modify Frame Structure*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN**

<b>Description</b>	<p>This command enables or disables VLAN type frames for the selected traffic stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Modify Frame Structure &gt; Vlan Tag</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN &lt;wsp&gt; &lt;Tgen&gt;, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Set the VLAN for the stream.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:VLAN 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN? 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN?</p>



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN?**

---

<b>Description</b>	<p>This query returns the status of VLAN type frames for the selected traffic stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Modify Frame Structure &gt; Vlan Tag</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of VLAN type frames.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:VLAN 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN? 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STacked</p>

---

## SCPI Command Reference

### *Modify Frame Structure*

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACked

---

<b>Description</b>	<p>This command sets the value of the Virtual Local Area Network (VLAN) stacked for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Modify Frame Structure &gt; Vlan Tag (dropdown)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACked <wsp> <Tgen>, <Stacked>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Stacked:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:VLAN 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:STAC 1, 3</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:STAC? 1</p> <p>Returns: 3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACked?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACked?**

---

<b>Description</b>	<p>This query returns the value of the VLAN stacked for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Modify Frame Structure &gt; Vlan Tag (dropdown)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACked? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Stacked>
<b>Response(s)</b>	<p><b>Stacked:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the VLAN stacked.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:VLAN 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:STAC 1, 3</p> <p>SOUR:DATA:TEL:ETH:STR:VLAN:STAC? 1</p> <p>Returns: 3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN:STACked</p>

---

## SCPI Command Reference

### Modify Frame Structure

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN

---

**Description** This command enables or disables the Virtual Local Area Network (VLAN) CONFig for frame parameter.

At \*RST condition, this value is set to OFF.

Navigation Path:Test Configurator>Setup>Services>MAC/IP/UDP>Modify frame structure>VLAN>VLAN tag

**Syntax** :SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN <wsp> <Service>, LTOR | RTOL, ON | OFF

**Parameter(s)** **Service:**  
The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the Service number 1 or 10.

**Direction:**

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: LTOR | RTOL

Selects the Direction.

(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)

LTOR: Local to Remote

RTOL: Remote to Local

**Set:**

The program data syntax for the third parameter is defined as a <Boolean Program Data> element.

The allowed elements for this parameter are: ON | OFF

Enables or disables the VLAN.

**Example(s)** SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON

**See Also** FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:RESUlt:SCOTest:BURSt:TEST?

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN?**

---

<b>Description</b>	<p>This command enables or disables the Virtual Local Area Network (VLAN) CONFig for frame parameter.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Test Configurator&gt;Setup&gt;Services&gt;MAC/IP/UDP&gt;Modify frame structure&gt;VLAN&gt;VLAN tag</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</code>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<code>&lt;Set&gt;</code>

---

## SCPI Command Reference

### *Modify Frame Structure*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN?**

<b>Response(s)</b>	<b>Set:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Virtual Local Area Network (VLAN) Configuration. 1, VLAN is enabled. 0, VLAN is disabled.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN? 1, LTOR Returns: 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam: ETHER?

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STACked**

<b>Description</b>	<p>This command selects the Virtual Local Area Network (VLAN) stacked for frame parameters for selected service and direction.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: EtherSAM&gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Vlan &gt; VLAN ID</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STACked <wsp> <Service>, LTOR   RTOL, <Stacked>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Stacked:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the VLAN stacked.</p>
<b>Example(s)</b>	<p>SOURCE:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON</p> <p>SOURCE:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC 1, LTOR,3</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN

## SCPI Command Reference

### Modify Frame Structure

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STACked?**

<b>Description</b>	<p>This query returns the Virtual Local Area Network (VLAN) stacked for frame parameters for selected service and selected direction.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: EtherSAM&gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; Vlan &gt; VLAN ID</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STACked? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Stacked&gt;</p>
<b>Response(s)</b>	<p><b>Stacked:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the VLAN stacked.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN 1, LTOR,ON</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC 1, LTOR,3</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN:STAC? 1, LTOR</p> <p>Returns: 3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:VLAN?</p>



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion**

---

<b>Description</b>	<p>This command selects the IP Version.</p> <p>At *RST condition, this value is set to IPV4.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; Modify Frame structure &gt; Global Option &gt; IP Version</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion <wsp>IPV6   IPV4
<b>Parameter(s)</b>	<p><b>IP Version:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: IPV6   IPV4</p> <p>Selects the IP version.</p> <p>IPV6</p> <p>IPV4</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:IPV IPV6</p> <p>SOUR:DATA:TEL:ETH:PORT:IPV?</p> <p>Returns: IPV6</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPV:ADDRess?

---

## SCPI Command Reference

### *Modify Frame Structure*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion?**

<b>Description</b>	<p>This query returns the IP Version.</p> <p>At *RST condition, this value is set to IPV4.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; Modify Frame structure &gt; Global Option &gt; IP Version</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion?
<b>Response Syntax</b>	<IP Version>
<b>Response(s)</b>	<p><b>IP Version:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the IP version.</p> <p>IPV4   IPV6</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:IPV IPV6</p> <p>SOUR:DATA:TEL:ETH:PORT:IPV?</p> <p>Returns: IPV6</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPV:MODE?

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS**

---

<b>Description</b>	<p>This command enables or disables the Multi Protocol Label Switching (MPLS) Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt; Modify Frame structure &gt; MPLS</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS <wsp> <Stream>, ON   OFF
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the stream from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the MPLS selection for selected stream</p>
<b>Example(s)</b>	<p>SOURCE:DATA:TELEcom:ETHernet:STReam:MPLS 1, on</p> <p>SOURCE:DATA:TELEcom:ETHernet:STReam:MPLS? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:HEADers

---

## SCPI Command Reference

### *Modify Frame Structure*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS?**

<b>Description</b>	<p>This query returns the status of Multi Protocol Label Switching (MPLS) Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test&gt; Traffic Gen and Mon/EtherSAM &gt; Test Configurator&gt; Set Up &gt; Streams&gt;MAC/IP/UDP &gt; Modify Frame structure &gt; MPLS</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS? <wsp> <Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the stream from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns MPLS status.</p> <p>1, MPLS is enabled.</p> <p>0, MPLS is disabled.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:STReam:MPLS 1, on</p> <p>SOURce:DATA:TELEcom:ETHernet:STReam:MPLS? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:HEADers?

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:HEADers**

---

<b>Description</b>	<p>This command Set the MPLS Header.</p> <p>At *RST condition, this value is set to 2.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt; MPLS &gt; MPLS Header</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:HEADers &lt;wsp&gt; &lt;Stream&gt;, &lt;Label Count&gt;</code>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the stream from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Label Count:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Label Count.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:MPLS:HEAD 1, 1 SOUR:DATA:TEL:ETH:STR:MPLS:HEAD? 1 Returns: 1</pre>
<b>See Also</b>	<code>SOURCE[1..n]:DATA:TELEcom:ELECtrical:STReam:MPLS:LABel</code>

---

## SCPI Command Reference

### *Modify Frame Structure*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:HEADers?**

---

<b>Description</b>	<p>This query returns the MPLS Header.</p> <p>At *RST condition, this value is set to 2.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Test Configurator&gt; Set Up &gt;Streams&gt;MAC/IP/UDP &gt; MPLS &gt; MPLS Header</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:HEADers? <wsp> <Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the stream from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Label Count (header)>
<b>Response(s)</b>	<p><b>Label Count (header):</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the label count(Header).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MPLS:HEAD 1, 1</p> <p>SOUR:DATA:TEL:ETH:STR:MPLS:HEAD? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSExp?

---

## TOS/DS Configuration

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS**

---

<b>Description</b>	<p>This command enables or disables the differentiated service status for the selected traffic stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS &lt;wsp&gt; &lt;Tgen&gt;, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable or disable the DS.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DS 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:DS? 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS?</p>

---

## SCPI Command Reference

### TOS/DS Configuration

---

**:SOURce[1..n]:DATA:TELecom:ETHernet:STReam:DS?**

<b>Description</b>	<p>This query returns the status of the differentiated service status for the selected traffic stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELecom:ETHernet:STReam:DS? &lt;wsp&gt; &lt;Tgen&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Set&gt;</p>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of differentiated service.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DS 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:DS? 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELecom:ETHernet:STReam:DS</p>

---



**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE**

<b>Description</b>	<p>This command sets the DSCP code points value for the selected traffic stream.</p> <p>At *RST condition, this value is set to CS0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; DS &gt; DSCP Code Points</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE <wsp> <Tgen>, CS0   CS1   CS2   CS3   CS4   CS5   CS6   CS7   AF11   AF12   AF13   AF21   AF22   AF23   AF31   AF32   AF33   AF41   AF42   AF43   EF   UCODE   DSCP51   DSCP54
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Code:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CS0   CS1   CS2   CS3   CS4   CS5   CS6   CS7   AF11   AF12   AF13   AF21   AF22   AF23   AF31   AF32   AF33   AF41   AF42   AF43   EF   UCODE   DSCP51   DSCP54</p> <p>Selects the code point for the differentiated services.</p> <p>UCODE: User Defined</p> <p>DSCP51: 51</p> <p>DSCP54: 54</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DS 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:DS:CODE 1, CS1</p> <p>SOUR:DATA:TEL:ETH:STR:DS:CODE? 1</p> <p>Returns: CS1</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS</p> <p>SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE?</p>

## SCPI Command Reference

### *TOS/DS Configuration*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE?**

<b>Description</b>	<p>This query returns the available DSCP code points value for the selected traffic stream.</p> <p>At *RST condition, this value is set to CS0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; DS &gt; DSCP Code Points</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE? &lt;wsp&gt; &lt;Tgen&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Code&gt;</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHerNet:STReam:DS:CODE?****Response(s)****Code:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Return the DS codepoint value.

CS0, selects CS0 as the DS code point.

CS1, selects CS1 as the DS code point.

CS2, selects CS2 as the DS code point.

CS3, selects CS3 as the DS code point.

CS4, selects CS4 as the DS code point.

CS5, selects CS5 as the DS code point.

CS6, selects CS6 as the DS code point.

CS7, selects CS7 as the DS code point.

AF11, selects AF11 as the DS code point.

AF12, selects AF12 as the DS code point.

AF13, selects AF13 as the DS code point.

AF21, selects AF21 as the DS code point.

AF22, selects AF22 as the DS code point.

AF23, selects AF23 as the DS code point.

AF31, selects AF31 as the DS code point.

AF32, selects AF32 as the DS code point.

AF33, selects AF33 as the DS code point.

AF41, selects AF41 as the DS code point.

AF42, selects AF42 as the DS code point.

AF43, selects AF43 as the DS code point.

EF, selects EF as the DS code point.

DSCP51, selects 51 as the DS code point.

DSCP54, selects 54 as the DS code point.

UCODE, selects User Defined as the DS code point.

---

## SCPI Command Reference

### *TOS/DS Configuration*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE?**

<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:DS 1, ON SOUR:DATA:TEL:ETH:STR:DS:CODE 1, CS1 SOUR:DATA:TEL:ETH:STR:DS:CODE? 1 Returns: CS1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:CODE

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:ECN**

<b>Description</b>	<p>This command sets the value of Explicit Congestion Notification (ECN) field for the selected traffic stream.</p> <p>At *RST condition, this value is set to 00 (Not-ECT).</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; DS &gt; ECN</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:ECN &lt;wsp&gt; &lt;Tgen&gt;, NECT   ECT1   ECT0   CE</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Ecn:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NECT   ECT1   ECT0   CE</p> <p>Sets the value for ECN (Explicit Congestion Notification).</p> <p>NECT: Not ECT (ECN Capable Transport)</p> <p>ECT1: ECT-1</p> <p>ECT0: ECT-0</p> <p>CE: CE (European Conformity)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DS 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:DS:ECN 1, NECT</p> <p>SOUR:DATA:TEL:ETH:STR:DS:ECN? 1</p> <p>Returns: NECT</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:ECN?</p>

## SCPI Command Reference

### TOS/DS Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:ECN?**

<b>Description</b>	<p>This query returns the value of the Explicit Congestion Notification (ECN) field for the selected traffic stream.</p> <p>At *RST condition, this value is set to 00 (Not-ECT).</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; DS &gt; ECN</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:ECN? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Ecn>
<b>Response(s)</b>	<p><b>Ecn:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the value of Explicit Congestion Notification (ECN).</p> <p>NECT, selects Not ECT (ECN Capable Transport) as the ECN value.</p> <p>ECT1, selects ECT-1 as the ECN value.</p> <p>ECT0, selects ECT-0 as the ECN value.</p> <p>CE, selects CE (European Conformity) as the ECN value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:DS 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:DS:ECN 1, NECT</p> <p>SOUR:DATA:TEL:ETH:STR:DS:ECN? 1</p> <p>Returns: NECT</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DS:ECN</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:BIT**

---

<b>Description</b>	<p>This command selects the reserved bit value of Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Reserved Bit</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:BIT <wsp> <Tgen>, <Bit>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Bit:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 0   1</p> <p>Selects the reserved bit value.</p> <p>0</p> <p>1</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:BIT 4,#B1</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:BIT? 4</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:BIT?

---

## SCPI Command Reference

### TOS/DS Configuration

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:BIT?

---

<b>Description</b>	<p>This query returns the reserved bit value of the Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Reserved Bit</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:BIT? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Bit>
<b>Response(s)</b>	<p><b>Bit:</b></p> <p>The response data syntax for this parameter is defined as a &lt;BINARY NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the reserved bit value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:BIT 4,#B1</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:BIT? 4</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:BIT

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:COST**

<b>Description</b>	<p>This command sets the monetary cost level of the Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Monetary Cost</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:COST <wsp> <Tgen>, NORMal   TLOW
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Cost:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NORMal   TLOW</p> <p>Selects the monetary cost level of the Type of Service.</p> <p>NORMal: Normal</p> <p>TLOW: Low</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:COST 1, TLOW</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:COST? 1</p> <p>Returns: TLOW</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:COST?

## SCPI Command Reference

### TOS/DS Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:COST?**

<b>Description</b>	<p>This query returns the monetary cost level of the Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Monetary Cost</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:COST? &lt;wsp&gt; &lt;Tgen&gt;</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<code>&lt;Cost&gt;</code>
<b>Response(s)</b>	<p><b>Cost:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the monetary cost level of the Type of Service.</p> <p>NORMal, selects Normal as the monetary cost level.</p> <p>TLOW, selects Low as the monetary cost level.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:TOS:COST 1, TLOW SOUR:DATA:TEL:ETH:STR:TOS:COST? 1 Returns: TLOW</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:COST</code>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DELAy**

<b>Description</b>	<p>This command sets the delay level of the Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Delay</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DELAy <wsp> <Tgen>, NORMal   TLOW
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Delay:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NORMal   TLOW</p> <p>Selects the delay level of the Type of Service.</p> <p>NORMal: Normal</p> <p>TLOW: Low</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:DEL 1, TLOW</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:DEL? 1</p> <p>Returns: TLOW</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DELAy?

---

## SCPI Command Reference

### TOS/DS Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DElay?**

<b>Description</b>	<p>This query returns the delay level of the Type of Service (TOS) for the selected traffic stream. At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Delay</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DElay? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Delay>
<b>Response(s)</b>	<p><b>Delay:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the delay level of the Type of Service.</p> <p>NORMal, selects Normal as the delay level.</p> <p>TLOW, selects Low as the delay level.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:DEL 1, TLOW</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:DEL? 1</p> <p>Returns: TLOW</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DElay

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECedence**

---

<b>Description</b>	<p>This command sets the precedence of the Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to ROUTine.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Precedence</p>
<b>Syntax</b>	<pre>:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECedence &lt;wsp&gt; &lt;Tgen&gt;, ROUTINE   PRIORITY   IMMEDIATE   FLASH   FOVERRIDE   CRITIC   ICONTROL   NCONTROL</pre>

---

## SCPI Command Reference

### TOS/DS Configuration

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECedence

<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: ROUTINE   PRIORITY   IMMEDIATE   FLASH   FOVERRIDE   CRITIC   ICONTROL   NCONTROL</p> <p>Selects the precedence of the Type of Service (TOS) for the selected traffic stream.</p> <p>ROUTINE: Routine.</p> <p>PRIORITY: Priority.</p> <p>IMMEDIATE: Immediate.</p> <p>FLASH: Flash.</p> <p>FOVERRIDE: Flash Override.</p> <p>CRITIC: Critic.</p> <p>ICONTROL: Internet Control.</p> <p>NCONTROL: Network Control.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:PREC 1, ROUT</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:PREC? 1</p> <p>Returns: ROUTINE</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECedence?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DELay</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECedence?**

<b>Description</b>	<p>This query returns the precedence of the Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to ROUTine.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Precedence</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECedence? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Precedence>
<b>Response(s)</b>	<p><b>Precedence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the precedence of the Type of Service.</p> <p>ROUTINE, Routine is selected as the TOS precedence.</p> <p>PRIORITY, Priority is selected as precedence.</p> <p>IMMEDIATE, Immediate is selected as precedence.</p> <p>FLASH, Flash is selected as precedence.</p> <p>FOVERRIDE, Flash Override is selected as precedence.</p> <p>CRITIC, Critic is selected as precedence.</p> <p>ICONTROL, Internet Control is selected as precedence.</p> <p>NCONTROL, Network Control is selected as precedence.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:PREC 1, ROUT</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:PREC? 1</p> <p>Returns: ROUTINE</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECedence</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:DELay</p>

## SCPI Command Reference

### TOS/DS Configuration

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:RELIability

<b>Description</b>	<p>This command sets the reliability level of the Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Reliability</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:RELIability <wsp> <Tgen>, NORMal   HIGH
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Reliability:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NORMal   HIGH</p> <p>Selects the reliability level of the Type of Service.</p> <p>NORMal: Normal</p> <p>HIGH: High</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:REL 1, HIGH</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:REL? 1</p> <p>Returns: HIGH</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:RELIability?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECedence</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:RELIability?**

---

<b>Description</b>	<p>This query returns the reliability level of the Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Reliability</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:RELIability? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Reliability>
<b>Response(s)</b>	<p><b>Reliability:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the reliability level of the Type of Service.</p> <p>NORMAL, selects the reliability level as Normal.</p> <p>HIGH, selects the reliability level as High.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:REL 1, HIGH</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:REL? 1</p> <p>Returns: HIGH</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:RELIability</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PREcedence</p>

---

## SCPI Command Reference

### TOS/DS Configuration

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:THROUGHput

<b>Description</b>	<p>This command sets the throughput level of the Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Throughput</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:THROUGHput <wsp> <Tgen>, NORMal   HIGH
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p> <p><b>Throughput:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NORMal   HIGH</p> <p>Selects the throughput level of the Type of Service.</p> <p>NORMal: Normal</p> <p>HIGH: High</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:THR 1, HIGH</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:THR? 1</p> <p>Returns: HIGH</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:THROUGHput?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PRECedence</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:THROUGHput?**

<b>Description</b>	<p>This query returns the throughput level of the Type of Service (TOS) for the selected traffic stream.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: BERT &gt; Test Setup &gt; Test Configurator &gt; MAC/IP/UDP &gt; IP TOS/DS &gt; TOS/DS Config &gt; TOS/DS &gt; TOS &gt; Throughput</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:THROUGHput? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM: the service from 1 to 10.</p> <p>Traffic Gen and Mon: the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Throughput>
<b>Response(s)</b>	<p><b>Throughput:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the throughput level of the Type of Service.</p> <p>NORMAL, Normal is selected as the throughput level.</p> <p>HIGH, High is selected as the throughput level.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TOS:THR 1, HIGH</p> <p>SOUR:DATA:TEL:ETH:STR:TOS:THR? 1</p> <p>Returns: HIGH</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:THROUGHput</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TOS:PREcedence</p>

## Configure Per Frame Size

---

:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ALL:FRAMe

---

<b>Description</b>	<p>This command enables or disables the All Frames RFC 2544 latency subtest.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC &gt; Configure Per Frame Size &gt; All Frames</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ALL:FRAMe <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Test:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables All Frames.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:ALL:FRAM ON</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:ALL:FRAM?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOsS:TTIME

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ALL:FRAMe?**

<b>Description</b>	<p>This query returns the status of 'All Frames' checkbox of RFC 2544 latency subtest.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC &gt; Configure Per Frame Size &gt; All Frames</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:ALL:FRAMe?
<b>Response Syntax</b>	<Get>
<b>Response(s)</b>	<p><b>Get:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of 'All Frames'.</p> <p>0, returns the status of checkbox as disabled.</p> <p>1, returns the status of checkbox as enabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:ALL:FRAM ON</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:ALL:FRAM?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOsS:TTIME?

---

## GFP-F/GFP-T

**:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:EXI**

<b>Description</b>	<p>This command sets the Extension Header Identifier.</p> <p>At *RST condition, this value is Null.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Setup &gt; GFP-F &gt; EXI</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:EXI &lt;wsp&gt;NULL   LINEAR</p>
<b>Parameter(s)</b>	<p><b>Value:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NULL   LINEAR</p> <p>Selects the Extension Header Identifier value</p> <p>NULL</p> <p>LINEAR</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:CONF:EXI NULL</p> <p>SOUR:DATA:TEL:GFP:CONF:EXI?</p> <p>Returns: NULL</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFiguration:TYPE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:EXI?**

---

<b>Description</b>	<p>This query returns the Extension Header Identifier.</p> <p>At *RST condition, this value is Null.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Setup &gt; GFP-F &gt; EXI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:EXI?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the status of Extension Header Identifier</p> <p>NULL, returns NULL as Extension Header Identifier</p> <p>LINEAR, returns LINEAR as Extension Header Identifier</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:CONF:EXI NULL</p> <p>SOUR:DATA:TEL:GFP:CONF:EXI?</p> <p>Returns: NULL</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFiguration:TYPE

---

## SCPI Command Reference

### GFP-F/GFP-T

---

**:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:TYPE**

<b>Description</b>	<p>This command sets the presence of expected payload FCS for Frames.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Setup &gt; GFP-F &gt; Type Header</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Setup &gt; GFP-T &gt; Type Header</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:TYPE &lt;wsp&gt;CDFRames   CMRFRames, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CDFRames   CMRFRames</p> <p>Selects the Client Frame Type.</p> <p>CDFRame: CDFRames</p> <p>CMFRames: CMFRames</p> <p><b>SET:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:CHAN:CONF:TYPE CDFR,ON</p> <p>SOUR:DATA:TEL:GFP:CHAN:CONF:TYPE? CDFR</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFiguration:CID?</p>



---

**:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:TYPE?**

---

<b>Description</b>	<p>This query returns the presence of expected payload FCS for Frames.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Setup &gt; GFP-F &gt; Type Header</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Setup &gt; GFP-T &gt; Type Header</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:TYPE? <wsp>CDFRames   CMRFRames
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CDFRames   CMRFRames</p> <p>Selects the Client Frame Type.</p> <p>CDFRames: CDFRames</p> <p>CMRFRames: CMRFRames</p>
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the presence of expected payload Gfp for Frames.</p> <p>1 -expected payload Gfp for Frames is enabled.</p> <p>0 -expected payload Gfp for Frames is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:CHAN:CONF:TYPE CDFR,ON</p> <p>SOUR:DATA:TEL:GFP:CHAN:CONF:TYPE? CDFR</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFiguration:CID

---

## SCPI Command Reference

### GFP-F/GFP-T

---

**:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:CID**

<b>Description</b>	<p>This command sets the Channel Id.</p> <p>At *RST condition, this value is 0.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Setup &gt; GFP-F &gt; Type Header &gt; EXI -(Linear) &gt; Extension Header &gt; CID</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:CID &lt;wsp&gt; &lt;Value&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Value:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects Channel Id between 0 to 255.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:CHAN:CONF:CID 230</p> <p>SOUR:DATA:TEL:GFP:CHAN:CONF:CID?</p> <p>Returns: 230</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:CID?**

<b>Description</b>	<p>This query returns the Channel Id.</p> <p>At *RST condition, this value is 0.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Setup &gt; GFP-F &gt; Type Header &gt; EXI -(Linear) &gt; Extension Header &gt; CID</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:CID?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified the CONFig is 0.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Channel used for reception.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:CHAN:CONF:CID 230</p> <p>SOUR:DATA:TEL:GFP:CHAN:CONF:CID?</p> <p>Returns: 230</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFig:TYPE

## Signal - Signal Configuration (SONET/SDH)

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:S:BITS:SSMessage**

---

<b>Description</b>	<p>This command selects the Bits 5-8 (Synchronization Status Message).</p> <p>At *RST condition, this value is set to QUNKNOWN0000.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Signal CONFig &gt; Synchronization Status Message(s1)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:S:BITS:SSMessage &lt;wsp&gt;QUNKNOWN0000   ST10001   RESERVED0010   RESERVED0011   TNC0100   RESERVED0101   RESERVED0110   ST20111   RESERVED1000   RESERVED1001   ST31010   RESERVED1011   SMC1100   PNO1101   ST3E1110   DUSYNCH1111</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:S:BITS:SSMessage**

<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: QUNKNOW0000   ST10001   RESERVED0010   RESERVED0011   TNC0100   RESERVED0101   RESERVED0110   ST20111   RESERVED1000   RESERVED1001   ST31010   RESERVED1011   SMC1100   PNO1101   ST3E1110   DUSYNCH1111</p> <p>Sets the Bits 5-8 (Synchronization Status Message).</p> <p>QUNKNOW0000: Quality Unknown (0000).</p> <p>ST10001: ST1 (0001).</p> <p>RESERVED0010: RESERVED(0010).</p> <p>RESERVED0011: Reserved (0011).</p> <p>TNC0100: TNC (0100).</p> <p>RESERVED0101: Reserved (0101).</p> <p>RESERVED0110: Reserved (0110).</p> <p>ST20111: ST2 (0111).</p> <p>RESERVED1000: Reserved (1000).</p> <p>RESERVED1001: Reserved (1001).</p> <p>ST31010: ST3 (1010).</p> <p>RESERVED1011: Reserved (1011).</p> <p>SMC1100: SMC (1100).</p> <p>PNO1101: PNO (1101).</p> <p>ST3E1110: ST3E (1110).</p> <p>DUSYNCH1111: Don't Use for Synchronization (1111).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:S:BITS:SSM QUNKNOW0000</p> <p>SOUR:DATA:TEL:SDHS:ADV:S:BITS:SSM?</p> <p>Returns: QUNKNOW0000</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:HOP:TYPE?</p>

## SCPI Command Reference

### *Signal - Signal Configuration (SONET/SDH)*

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:S:BITS:SSMessage?**

<b>Description</b>	This query returns Bits 5-8 (Synchronization Status Message). At *RST condition, this value is set to QUNKNOWN0000. Navigation Path: Test Setup > OTN SONET/SDH BERT OR SONET/SDH BERT > Test Configurator > Setup > Signal CONFig > Synchronization Status Message(s1)
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:S:BITS:SSMessage?
<b>Response Syntax</b>	<Type>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:S:BITS:SSMessage?**

**Response(s)**

**Type:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Sets the Bits 5-8 (Synchronization Status Message).

QUNKNOWN0000 - Quality Unknown (0000) is selected.

ST10001 - ST1 (0001) is selected.

RESERVED0001 -RESERVED (0010) is selected.

RESERVED0011 - Reserved (0011) is selected.

TNC0100 - TNC (0100) is selected.

RESERVED0101- Reserved (0101) is selected.

RESERVED0110- Reserved (0110) is selected.

ST20111 - ST2 (0111) is selected.

RESERVED1000 - RESERVED (1000) is selected.

RESERVED1001- Reserved (1001) is selected.

ST31010 - ST3 (1010) is selected.

RESERVED1010 - RESERVED (SEC) (1011) is selected.

SMC1100 - SMC (1100) is selected.

ST3E1101 - ST3E (1101) is selected.

PNO1110 - PNO (1110) is selected.

DUSYNCH1111- Don't Use for Synchronization (1111) is selected.

**Example(s)**

SOUR:DATA:TEL:SDHS:ADV:S:BITS:SSM QUNKNOWN0000

SOUR:DATA:TEL:SDHS:ADV:S:BITS:SSM?

Returns: QUNKNOWN0000

**See Also**

SOURce[1..n]:DATA:TELEcom:HOP:TYPE

## SCPI Command Reference

### *Signal - Signal Configuration (SONET/SDH)*

---

---

#### :SOURce[1..n]:DATA:TELEcom:BACKground:COMPUtation

---

<b>Description</b>	<p>This command selects of the computation method.</p> <p>At *RST condition, this value set to MIONLY.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Signal CONFig &gt; REI-L Computation Method</p> <p>Appears in GUI when OC-192 or STM-64 is selected in Embedded Sonet Sdh field in Modify structure pop up.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:BACKground:COMPUtation <wsp>MIONLY   M0M1
<b>Parameter(s)</b>	<p><b>COMPUtation:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MIONLY   M0M1</p> <p>Selects the payload computation method.</p> <p>MIONLY: M1</p> <p>M0M1: M1 and M0</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:BACK:COMP M0M1</p> <p>SOUR:DATA:TEL:BACK:COMP?</p> <p>Returns: M0M1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP?

---



---

**:SOURce[1..n]:DATA:TELEcom:BACKground:COMPuTation?**

---

<b>Description</b>	<p>This query returns the computation method.</p> <p>At *RST condition, this value set to MIONLY.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Signal CONFig &gt; REI-L Computation Method</p> <p>Appears in GUI when OC-192 or STM-64 is selected in Embedded Sonet Sdh field in Modify structure pop up.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:BACKground:COMPuTation?
<b>Response Syntax</b>	<Computation>
<b>Response(s)</b>	<p><b>Computation:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the computation method.</p> <p>MIONLY, M1 as computation method is selected.</p> <p>M0M1, M1 and M0 as computation method is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:BACK:COMP M0M1</p> <p>SOUR:DATA:TEL:BACK:COMP?</p> <p>Returns: M0M1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP

---

## SCPI Command Reference

### Signal - Signal Configuration (SONET/SDH)

---

**:SOURce[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP**

<b>Description</b>	<p>This command selects the High Order Path (HOP) background traffic.</p> <p>At *RST condition, this value set to EQUIPPED1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Signal CONFig &gt; Background Traffic</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP &lt;wsp&gt;AIS   UNEQUIPPED1   EQUIPPED1</p>
<b>Parameter(s)</b>	<p><b>Background Traffic:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   UNEQUIPPED1   EQUIPPED1</p> <p>Selects the High Order Path (HOP) background traffic.</p> <p>AIS: Alarm Indication Signal</p> <p>UNEQUIPPED1: Unequipped</p> <p>EQUIPPED1: Equipped</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:BACK:SDHS:HOP EQUIPPED1</p> <p>SOUR:DATA:TEL:BACK:SDHS:HOP?</p> <p>Returns: EQUIPPED1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:BACKground:COMPutation</p>

---

**:SOURce[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP?**

---

<b>Description</b>	<p>This query returns the High Order Path (HOP) background traffic.</p> <p>At *RST condition, this value set to EQUIPPED1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Signal CONFig &gt; Background Traffic</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP?
<b>Response Syntax</b>	<Background traffic>
<b>Response(s)</b>	<p><b>Background traffic:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the High Order Path (HOP) background traffic.</p> <p>AIS, Alarm Indication Signal (AIS) as background traffic is selected.</p> <p>UNEQUIPPED1, Unequipped as background traffic is selected.</p> <p>EQUIPPED1, Equipped as background traffic is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:BACK:SDHS:HOP EQUIPPED1</p> <p>SOUR:DATA:TEL:BACK:SDHS:HOP?</p> <p>Returns: EQUIPPED1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:BACKground:COMPutation?

---

## SCPI Command Reference

### *Signal - Signal Configuration (SONET/SDH)*

---

**:SOURce[1..n]:DATA:TELEcom:BACKground:BULK**

<b>Description</b>	<p>This command enables or disables the Bulk Filled Overwrite Enable feature.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Signal CONFig &gt; Overwrite Fixed Stuff</p> <p>Appears in GUI when STS-1 is selected in Sonet Multiplexing field in Modify structure pop up.</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:BACKground:BULK &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Overwrite Fixed Stuff feature.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:BACK:BULK ON</p> <p>SOUR:DATA:TEL:BACK:BULK?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP</p>

---

**:SOURce[1..n]:DATA:TELEcom:BACKground:BULK?**

<b>Description</b>	<p>This query returns the status of the Bulk Filled Overwrite Enable feature.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Signal CONFig &gt; Overwrite Fixed Stuff</p> <p>Appears in GUI when STS-1 is selected in Sonet Multiplexing field in Modify structure pop up.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:BACKground:BULK?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Bulk Filled Overwrite Enable feature.</p> <p>1 - Bulk overwrite fixed Stuff Enabled.</p> <p>0 - Bulk overwrite fixed Stuff Disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:BACK:BULK ON</p> <p>SOUR:DATA:TEL:BACK:BULK?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:BACKground:SDHSonet:HOP?

## SCPI Command Reference

### *Signal - Signal Configuration (SONET/SDH)*

---

**:SOURCE[1..n]:DATA:TELECOM:POSITION**

<b>Description</b>	<p>This command sets the mapping position.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT &gt;Test Configurator&gt;HOP TYPE&gt;Setup&gt;Signal&gt;Timeslot</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELECOM:POSITION &lt;wsp&gt;HOPPosition   LOPPosition, &lt;Position1&gt;, &lt;Position2&gt;, &lt;Position3&gt;, &lt;Position4&gt;, &lt;Position5&gt;</p>

---

:SOURCE[1..n]:DATA:TELEcom:POSITION

<b>Parameter(s)</b>	<p><b>Positionid:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: HOPPosition   LOPPosition</p> <p>Selects the mapping position.</p> <p>HOPPosition: HOP (High Order Path)</p> <p>LOPPosition: LOP (Low Order Path)</p> <p><b>Position1:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the mapping position values.</p> <p><b>Position2:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p><b>Position3:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the mapping position values.</p> <p><b>Position4:</b></p> <p>The program data syntax for the fifth parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the mapping position values.</p> <p><b>Position5:</b></p> <p>The program data syntax for the sixth parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the mapping position values.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:POS HOPP,1, 1, 1, 3,1</p> <p>SOUR:DATA:TEL:POS? HOPP</p> <p>Returns: (1, 1, 1, 3,1)</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:HOP:TYPE</p>

## SCPI Command Reference

### Signal - Signal Configuration (SONET/SDH)

---

**:SOURce[1..n]:DATA:TELEcom:POSition?**

<b>Description</b>	<p>This query returns the mapping position.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt;Test Configurator&gt;HOP TYPE&gt;Setup&gt;Signal&gt;Timeslot</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:POSition? <wsp>HOPPosition   LOPPosition
<b>Parameter(s)</b>	<p><b>Positionid:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: HOPPosition   LOPPosition</p> <p>Selects the mapping position.</p> <p>HOPPosition: HOP (High Order Path)</p>
<b>Response Syntax</b>	<Position>
<b>Response(s)</b>	<p><b>Position:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the mapping positions.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:POS HOPP,1, 1, 1, 3,1</p> <p>SOUR:DATA:TEL:POS? HOPP</p> <p>Returns: (1, 1, 1, 3,1)</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:HOP:TYPE?

---



---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TCUNeq:ENABle**

---

<b>Description</b>	<p>This command enables or disables the TCM Unique for Higher rates.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;STS&gt;TC-UNEQ-P</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;AU&gt;HPTC-UNEQ</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TCUNeq:ENABle <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the HOP TCM UNEQ.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:ALAR:HOP:TCM:TCUNeq:ENAB ON</p> <p>SENS:DATA:TEL:SDHS:ALAR:HOP:TCM:TCUNeq:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LOP:TCM:TCUNeq:ENABle?

---

## SCPI Command Reference

### Signal - Signal Configuration (SONET/SDH)

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TCUNeq:ENABle?**

<b>Description</b>	<p>This query returns the status of TCM Unique for Higher rates.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;STS&gt;TC-UNEQ-P</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;AU&gt;HPTC-UNEQ</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TCUNeq:ENABle?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of TCM UNEQ for Higher Rates.</p> <p>1, TCM UNEQ is enabled.</p> <p>0, TCM UNEQ is disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:ALAR:HOP:TCM:TCUNeq:ENAB ON</p> <p>SENS:DATA:TEL:SDHS:ALAR:HOP:TCM:TCUNeq:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LOP:TCM:TCUNeq:ENABle

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABle**

<b>Description</b>	<p>This command enables or disables the TCM for Higher rates.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;STS&gt;TCM</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;AU&gt;TCM</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABle <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the HOP TCM.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:CONF:TCM:ENAB ON</p> <p>SOUR:DATA:TEL:SDHS:HOP:CONF:TCM:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:CONFig:TCM:ENABle?

## SCPI Command Reference

### Signal - Signal Configuration (SONET/SDH)

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABLE?**

---

<b>Description</b>	<p>This query returns the status of TCM for Higher rates.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;STS&gt;TCM</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;AU&gt;TCM</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABLE?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of TCM for Higher Rates.</p> <p>1, TCM is enabled.</p> <p>0, TCM is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:CONF:TCM:ENAB ON</p> <p>SOUR:DATA:TEL:SDHS:HOP:CONF:TCM:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:CONFig:TCM:ENABLE

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABle**

<b>Description</b>	<p>This command enables or disables the TCM for Higher rates.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;STS&gt;TCM</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;AU&gt;TCM</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:CONFig:TCM:ENABle <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the HOP TCM.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:HOP:CONF:TCM:ENAB ON</p> <p>SENS:DATA:TEL:SDHS:HOP:CONF:TCM:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:LOP:CONFig:TCM:ENABle?

---

## SCPI Command Reference

### Signal - Signal Configuration (SONET/SDH)

---

**:SENSe[1..n]:DATA:TELeom:SDHSonet:HOP:CONFig:TCM:ENABle?**

<b>Description</b>	<p>This query returns the status of TCM for Higher rates.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;STS&gt;TCM</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Signal Configuration&gt;AU&gt;TCM</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:SDHSonet:HOP:CONFig:TCM:ENABle?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of TCM for Higher Rates.</p> <p>1, TCM is enabled.</p> <p>0, TCM is disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:HOP:CONF:TCM:ENAB ON</p> <p>SENS:DATA:TEL:SDHS:HOP:CONF:TCM:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:SDHSonet:LOP:CONFig:TCM:ENABle

## Traces (SONET/SDH)

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern**

---

<b>Description</b>	<p>This command sets the J1 trace format for Path Overhead.</p> <p>At *RST condition, this value is set to 1BYTE.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;Traces&gt;SONET&gt;STS Path (J1) &gt; Format</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;Traces&gt;SDH&gt;AU Path (J1) &gt; Format</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern &lt;wsp&gt;1BYTE   B16   B64</p>
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 1BYTE   B16   B64</p> <p>Sets the format for J1 trace for Path overhead.</p> <p>1BYTE: the 1 bytes format.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:PATH:OVER:J1:PATT B16</p> <p>SOUR:DATA:TEL:SDHS:PATH:OVER:J1:PATT?</p> <p>Returns: B16</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATTern?</p>

---

## SCPI Command Reference

### Traces (SONET/SDH)

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern?**

<b>Description</b>	<p>This query returns the J1 trace format for Path Overhead.</p> <p>At *RST condition, this value is set to 1BYTE.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; Traces &gt; SONET &gt; STS Path (J1) &gt; Format</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; Traces &gt; SDH &gt; AU Path (J1) &gt; Format</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern?
<b>Response Syntax</b>	<Pattern>
<b>Response(s)</b>	<p><b>Pattern:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the format for J1 trace for Path Overhead.</p> <p>B16, 16 bytes format is selected.</p> <p>B64, 64 bytes format is selected.</p> <p>1BYTE, 1 byte format is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:PATH:OVER:J1:PATT B16</p> <p>SOUR:DATA:TEL:SDHS:PATH:OVER:J1:PATT?</p> <p>Returns: B16</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATTern

---



---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern:B**

<b>Description</b>	<p>This command sets the J0 trace value/message to be generated in selected format for Section Overhead.</p> <p>At *RST condition, this value is set to ""01"".</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; Traces &gt; SONET &gt; STS Path (J1) &gt; Generated</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; Traces &gt; SDH &gt; AU Path (J1) &gt; Generated</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern:B &lt;wsp&gt;B16   B64   1BYTE, &lt;Message&gt;</p>
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64   1BYTE</p> <p>Sets the format for J1 trace for Path overhead.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p> <p>1BYTE: the 1BYTE byte format.</p> <p><b>Message:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the 16 or 64 byte format for J1 trace.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:PATH:OVER:J1:PATT:B B16,"RRWEREW"</p> <p>SOUR:DATA:TEL:SDHS:PATH:OVER:J1:PATT:B? B16</p> <p>Returns: "RRWEREW"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATTern:B?</p>

---

## SCPI Command Reference

### Traces (SONET/SDH)

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern:B?**

<b>Description</b>	<p>This query returns the J0 trace value/message to be generated in selected format for Section Overhead.</p> <p>At *RST condition, this value is set to ""01"".</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;Traces&gt;SONET&gt;STS Path (J1)&gt; Generated</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;Traces&gt;SDH&gt;AU Path (J1)&gt; Generated</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern:B? &lt;wsp&gt;B16   B64   1BYTE</p>
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64   1BYTE</p> <p>Sets the format for J1 trace for Path overhead.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p> <p>1BYTE: the 1BYTE byte format.</p>
<b>Response Syntax</b>	<p>&lt;Message&gt;</p>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the J1 trace value/message for Path Overhead.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:PATH:OVER:J1:PATT:B B16,"RRWEREW"</p> <p>SOUR:DATA:TEL:SDHS:PATH:OVER:J1:PATT:B? B16</p> <p>Returns: "RRWEREW"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATTern:B</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATTern**

---

<b>Description</b>	<p>This command sets the J0 trace format for Section Overhead.</p> <p>At *RST condition, this value is set to 1BYTE.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SONET &gt; Traces &gt; Section (J0) &gt; Format</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SDH &gt; Traces &gt; RS (J0) &gt; Format</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATTern &lt;wsp&gt;B16   B64   1BYTE</pre>
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64   1BYTE</p> <p>Sets the format for J0 trace for Section overhead.</p> <p>1BYTE: the 1 bytes format.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:SECT:OVER:J100:PATT B16 SOUR:DATA:TEL:SDHS:SECT:OVER:J100:PATT? Returns: B16</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern?</pre>

---

## SCPI Command Reference

### Traces (SONET/SDH)

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATtern?**

<b>Description</b>	<p>This query returns the J0 trace format for Section Overhead.</p> <p>At *RST condition, this value is set to 1BYTE.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SONET&gt;Traces&gt;Section (J0) &gt; Format</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SDH&gt;Traces&gt;RS (J0) &gt; Format</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATtern?
<b>Response Syntax</b>	<Pattern>
<b>Response(s)</b>	<p><b>Pattern:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the format for J0 trace for Section Overhead.</p> <p>B16, 16 bytes format is selected.</p> <p>B64, 64 bytes format is selected.</p> <p>1BYTE, 1 byte format is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:SECT:OVER:J100:PATT B16</p> <p>SOUR:DATA:TEL:SDHS:SECT:OVER:J100:PATT?</p> <p>Returns: B16</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATtern

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTION:OVERhead:J[1..n]:PATTern:B**

---

<b>Description</b>	<p>This command sets the 16 or 64 bytes message of the J0 trace for Section Overhead.</p> <p>At *RST condition, this value is set to 01.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SONET &gt; Traces &gt; Section (J0) &gt; Generated</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SDH &gt; Traces &gt; RS (J0) &gt; Generated</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTION:OVERhead:J[1..n]:PATTern:B &lt;wsp&gt;B16   B64   1BYTE, &lt;Message&gt;</p>
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64   1BYTE</p> <p>Sets the format for J0 trace for Section overhead.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p> <p>1BYTE: the 1 byte format.</p> <p><b>Message:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the 16 or 64 byte format for J0 trace.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:SECT:OVER:J100:PATT:B B16,"RRWEREW"</p> <p>SOUR:DATA:TEL:SDHS:SECT:OVER:J100:PATT:B? B16</p> <p>Returns: "RRWEREW"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATTern:B?</p>

---

## SCPI Command Reference

### Traces (SONET/SDH)

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATtern:B?**

<b>Description</b>	<p>This query returns the 64 bytes message of the J1/J2 trace for Section Overhead.</p> <p>At *RST condition, this value is set to 01.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SONET &gt; Traces &gt; Section (J0) &gt; Generated</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SDH &gt; Traces &gt; RS (J0) &gt; Generated</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:J[1..n]:PATtern:B? &lt;wsp&gt;B16   B64   1BYTE</p>
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64   1BYTE</p> <p>Sets the format for J0 trace for Section overhead.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p> <p>1BYTE: the 1 byte format.</p>
<b>Response Syntax</b>	<p>&lt;Message&gt;</p>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the J0 trace value/message for Section Overhead.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:SECT:OVER:J100:PATT:B B16,"RRWEREW"</p> <p>SOUR:DATA:TEL:SDHS:SECT:OVER:J100:PATT:B? B16</p> <p>Returns: "RRWEREW"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:J[1..n]:PATtern:B?</p>

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM**

---

<b>Description</b>	<p>This command enables or disables the Trace Identifier Mismatch (TIM) for the expected message defined for Path Overhead.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SONET &gt; Traces &gt; TIM-P</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SONET &gt; Traces &gt; TIM-P</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SDH &gt; Traces &gt; HP-TIM</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SDH &gt; Traces &gt; HP-TIM</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for Path overhead.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM ON</p> <p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM?

---

## SCPI Command Reference

### Traces (SONET/SDH)

---

#### :SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM?

<b>Description</b>	<p>This query returns the status of Trace Identifier Mismatch (TIM) for the expected message defined for Path Overhead.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SONET&gt;Traces &gt;TIM-P</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SONET&gt;Traces &gt;TIM-P</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SDH&gt;Traces &gt;HP-TIM</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SDH&gt;Traces &gt;HP-TIM</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Trace Identifier Mismatch (TIM) for Path Overhead.</p> <p>1 - TIM enabled.</p> <p>0 - TIM disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM ON</p> <p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM



**:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern**

<b>Description</b>	<p>This command sets the Trace Identifier Mismatch value of J1 in 16 or 64 byte format for Path Overhead.</p> <p>At *RST condition, this value is set to B16.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SONET &gt; Traces &gt; TIM-P &gt; Format</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SONET &gt; Traces &gt; TIM-P &gt; Format</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SDH &gt; Traces &gt; HP-TIM &gt; Format</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SDH &gt; Traces &gt; HP-TIM &gt; Format</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern <wsp>B16   B64
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64</p> <p>Sets the format expected in the J1 trace for Path overhead.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM:PATT B16</p> <p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM:PATT?</p> <p>Returns: B16</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern?

## SCPI Command Reference

### Traces (SONET/SDH)

---

---

#### :SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern?

---

<b>Description</b>	<p>This query returns the format expected in the J1 trace for Path Overhead.</p> <p>At *RST condition, this value is set to B16.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SONET&gt;Traces&gt;TIM-P</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SONET&gt;Traces&gt;TIM-P</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SDH&gt;Traces&gt;HP-TIM</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SDH&gt;Traces&gt;HP-TIM</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern?
<b>Response Syntax</b>	<Pattern>
<b>Response(s)</b>	<p><b>Pattern:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the format expected in the J1 trace for Path Overhead.</p> <p>B16, 16 bytes format is selected.</p> <p>B64, 64 bytes format is selected.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM:PATT B16</p> <p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM:PATT?</p> <p>Returns: B16</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern:B**

<b>Description</b>	<p>This command sets the expected message for 16 OR 64 bytes format of J1 trace for Path Overhead.</p> <p>At *RST condition, this value is set to ""EXFO SONET/SDH"".</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SONET&gt;Traces&gt;TIM-P &gt; Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SONET&gt;Traces&gt;TIM-P &gt; Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SDH&gt;Traces&gt;HP-TIM &gt; Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SDH&gt;Traces&gt;HP-TIM &gt; Expected</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern:B &lt;wsp&gt;B16   B64, &lt;Message&gt;</p>
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64</p> <p>Sets the format expected in the J1 trace for Path overhead.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p> <p><b>Message:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for 16 or 64 bytes format for Path overhead.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM:PATT:B B16,tyghjg</p> <p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM:PATT:B? B16</p> <p>Returns: "tyghjg"</p>
<b>See Also</b>	<p>SENSE[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern:B?</p>

## SCPI Command Reference

### Traces (SONET/SDH)

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern:B?**

<b>Description</b>	<p>This query returns the expected message for 16 or 64 bytes format of J1 trace for Path Overhead.</p> <p>At *RST condition, this value is set to ""EXFO SONET/SDH"".</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SONET&gt;Traces&gt;TIM-P &gt; Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SONET&gt;Traces&gt;TIM-P &gt; Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SDH&gt;Traces&gt;HP-TIM &gt; Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SDH&gt;Traces&gt;HP-TIM &gt; Expected</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern:B? &lt;wsp&gt;B16   B64</code>
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64</p> <p>Sets the format expected in the J1 trace for Path overhead.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p>
<b>Response Syntax</b>	<code>&lt;Message&gt;</code>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for 16 or 64 bytes format for Path Overhead.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM:PATT:B B16, tyghjg</p> <p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM:PATT:B? B16</p> <p>Returns: "tyghjg"</p>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern:B</code>

**:SENSe[1..n]:DATA:TELecom:SDHSonet:SECTion:OVERhead:TIM**

<b>Description</b>	<p>This command enables or disables the Trace Identifier Mismatch (TIM) for the expected message defined for Section Overhead.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SONET &gt; Traces &gt; TIM-S</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SONET &gt; Traces &gt; TIM-S</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SDH &gt; Traces &gt; RS-TIM</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SDH &gt; Traces &gt; RS-TIM</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:SDHSonet:SECTion:OVERhead:TIM <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for Section overhead.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:SECT:OVER:TIM ON</p> <p>SENS:DATA:TEL:SDHS:SECT:OVER:TIM?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:SDHSonet:PATH:OVERhead:TIM?

## SCPI Command Reference

### Traces (SONET/SDH)

---

#### :SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM?

**Description** This query returns the status of Trace Identifier Mismatch (TIM) for the expected message defined for Section Overhead.

At \*RST condition, this value is set to OFF.

Navigation Path: Test Setup > OTN SONET/SDH BERT OR SONET/SDH BERT > Test Configurator > Setup > Traces/Labels > SONET > Traces > TIM-S

Navigation Path: Test Setup > OTN SONET/SDH BERT OR SONET/SDH BERT > Test Configurator > Results > Traces > SONET > Traces > TIM-S

Navigation Path: Test Setup > OTN SONET/SDH BERT OR SONET/SDH BERT > Test Configurator > Setup > Traces/Labels > SDH > Traces > RS-TIM

Navigation Path: Test Setup > OTN SONET/SDH BERT OR SONET/SDH BERT > Test Configurator > Results > Traces > SDH > Traces > RS-TIM

**Syntax** :SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM?

**Response Syntax** <Set>

**Response(s)** Set:

The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of Trace Identifier Mismatch (TIM) for Section Overhead.

1 - TIM is Enabled

0 - TIM is Disabled

**Example(s)** SENS:DATA:TEL:SDHS:SECT:OVER:TIM ON

SENS:DATA:TEL:SDHS:SECT:OVER:TIM?

Returns: 1

**See Also** SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM

---

**:SENSe[1..n]:DATA:TELecom:SDHSonet:SECTion:OVERhead:TIM:PATtern**

<b>Description</b>	<p>This command sets the format expected in the J0 trace for Section Overhead.</p> <p>At *RST condition, this value is set to B16.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SONET &gt; Traces &gt; TIM-S &gt; Format</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SONET &gt; Traces &gt; TIM-S &gt; Format</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; SDH &gt; Traces &gt; RS-TIM &gt; Format</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SDH &gt; Traces &gt; RS-TIM &gt; Format</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:SDHSonet:SECTion:OVERhead:TIM:PATtern <wsp>B16   B64
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64</p> <p>Sets the format expected in the J0 trace for Section overhead.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:SECT:OVER:TIM:PATT B16</p> <p>SENS:DATA:TEL:SDHS:SECT:OVER:TIM:PATT?</p> <p>Returns: B16</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:SDHSonet:PATH:OVERhead:TIM:PATtern?

## SCPI Command Reference

### Traces (SONET/SDH)

---

---

#### :SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern?

---

<b>Description</b>	<p>This query returns the format expected in the J0 trace for Section Overhead.</p> <p>At *RST condition, this value is set to B16.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SONET&gt;Traces&gt;TIM-S&gt;Format</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SONET&gt;Traces&gt;TIM-S&gt;Format</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SDH&gt;Traces&gt;RS-TIM&gt;Format</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SDH&gt;Traces&gt;RS-TIM&gt;Format</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern?
<b>Response Syntax</b>	<Pattern>
<b>Response(s)</b>	<p><b>Pattern:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the format expected in the J0 trace for Section Overhead.</p> <p>B16, 16 bytes format is selected.</p> <p>B64, 64 bytes format is selected.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:SECT:OVER:TIM:PATT B16</p> <p>SENS:DATA:TEL:SDHS:SECT:OVER:TIM:PATT?</p> <p>Returns: B16</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern

---



**:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern:B**

<b>Description</b>	<p>This command sets the expected message for 16 or 64 bytes format of J0 trace for Section Overhead.</p> <p>At *RST condition, this value is set to ""EXFO SONET/SDH"".</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SONET&gt;Traces&gt; &gt; TIM-S &gt; Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SONET&gt;Traces&gt; TIM-S &gt; Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SDH&gt;Traces&gt; RS-TIM &gt;Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SDH&gt;Traces&gt; RS-TIM &gt;Expected</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern:B &lt;wsp&gt;B16   B64, &lt;Message&gt;</p>
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64</p> <p>Sets the format expected in the J0 trace for Section overhead.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p> <p><b>Message:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for 16 or 64 bytes format for Section overhead.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:SECT:OVER:TIM:PATT:B B16, tyghjg</p> <p>SENS:DATA:TEL:SDHS:SECT:OVER:TIM:PATT:B? B16</p> <p>Returns: "tyghjg"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern:B?</p>

## SCPI Command Reference

### Traces (SONET/SDH)

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern:B?**

<b>Description</b>	<p>This query returns the expected message for 16 or 64 bytes format of J0 trace for Section Overhead.</p> <p>At *RST condition, this value is set to ""EXFO SONET/SDH"".</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SONET&gt;Traces&gt; TIM-S &gt;Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SONET&gt;Traces&gt; TIM-S &gt;Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;SDH&gt;Traces &gt; RS-TIM &gt;Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;SDH&gt;Traces &gt; RS-TIM &gt;Expected</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern:B? &lt;wsp&gt;B16   B64</p>
<b>Parameter(s)</b>	<p><b>Pattern:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B16   B64</p> <p>Sets the format expected in the J0 trace for Section overhead.</p> <p>B16: the 16 bytes format.</p> <p>B64: the 64 bytes format.</p>
<b>Response Syntax</b>	<p>&lt;Message&gt;</p>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected message for 16 or 64 bytes format for Section Overhead.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:SECT:OVER:TIM:PATT:B B16, tyghjg</p> <p>SENS:DATA:TEL:SDHS:SECT:OVER:TIM:PATT:B? B16</p> <p>Returns: "tyghjg"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:PATtern:B</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:MESSAge**

---

<b>Description</b>	<p>This command sets the TCM Access Point Identifier message High Order Path (HOP). At *RST condition, the configuration is set to a device-dependent value.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces &gt;TCM Access Point Identifier&gt; STS Path&gt;Generated</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;TCM Access Point Identifier&gt; AU Path&gt;Generated</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:MESSAge <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the 16 or 64 byte format for TCM Access Point Identifier message.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:TCAP:N1:MESSAge tcmmessage</p> <p>SOUR:DATA:TEL:SDHS:HOP:TCAP:N1:MESSAge?</p> <p>Returns: "tcmmessage"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:PATtern:B16?

---

## SCPI Command Reference

### Traces (SONET/SDH)

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:MESSAge?**

<b>Description</b>	<p>This query returns the TCM Access Point Identifier message High Order Path (HOP). At *RST condition, the configuration is set to a device-dependent value.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces &gt;TCM Access Point Identifier&gt; STS Path&gt;Generated</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;TCM Access Point Identifier&gt; AU Path&gt;Generated</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:MESSAge?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the 16 or 64 byte format TCM Access Point Identifier message for Path Overhead.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:TCAP:N1:MESSAge tcmmessage</p> <p>SOUR:DATA:TEL:SDHS:HOP:TCAP:N1:MESSAge?</p> <p>Returns: "tcmmessage"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:PATtern:B16

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:EXPEcted**

<b>Description</b>	<p>This command sets the TCM Access Point Identifier expected message.</p> <p>At *RST condition, the configuration is set to a device-dependent value.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces &gt;TCM Access Point Identifier&gt; STS Path&gt;Expected</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;TCM Access Point Identifier&gt;AUPath&gt; Expected</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:EXPEcted <wsp> <Message>
<b>Parameter(s)</b>	<p><b>Message:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the expected message for 16 or 64 bytes format for Path overhead.</p>
<b>Example(s)</b>	<p>SENSe:DATA:TEL:SDHS:HOP:TCAP:EXPEcted tcmmessage</p> <p>SENSe:DATA:TEL:SDHS:HOP:TCAP:EXPEcted?</p> <p>Returns: "tcmmessage"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern?

---

## SCPI Command Reference

Traces (SONET/SDH)

---

### :SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:EXPeCted?

<b>Description</b>	<p>This command sets the TCM Access Point Identifier expected message High Order Path (HOP).</p> <p>At *RST condition, the configuration is set to a device-dependent value.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces &gt;TCM Access Point Identifier&gt; STS Path&gt;Expected</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;TCM Access Point Identifier&gt;AUPath&gt; Expected</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:EXPeCted?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected TCM Access Point Identifier message for Path Overhead.</p>
<b>Example(s)</b>	<p>SENSe:DATA:TEL:SDHS:HOP:TCAP:EXPeCted tcmmessage</p> <p>SENSe:DATA:TEL:SDHS:HOP:TCAP:EXPeCted?</p> <p>Returns: "tcmmessage"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:PATtern

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:TCTim**

<b>Description</b>	<p>This command enables or disables the TCMIM status for TCM Access Point Identifier High Order Path (HOP).</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;TCM Access Point Identifier&gt;HPTC-TIM</p> <p>Navigation Path: ""Test Setup &gt; SONET/SDH BERT&gt; Test Configurator &gt; Results &gt; Traces &gt; DSn/PDH &gt; TCM Access Point Identifier&gt;HPTC-TIM</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:TCTim <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the status of Trace Identifier Mismatch (TIM).</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:HOP:TCAP:TCTim ON</p> <p>SENS:DATA:TEL:SDHS:HOP:TCAP:TCTim?</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM?

## SCPI Command Reference

### Traces (SONET/SDH)

---

---

#### :SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:TCTim?

---

<b>Description</b>	<p>This query returns the the TCMIM status for TCM Access Point Identifier for High Order Path (HOP).</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces&gt;TCM Access Point Identifier&gt;HPTC-TIM</p> <p>Navigation Path: ""Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Results &gt; Traces&gt;TCM Access Point Identifier&gt;HPTC-TIM</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:TCTim?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Trace Identifier Mismatch (TIM).</p> <p>1, TCM Access Point Identifier is enabled.</p> <p>0, TCM Access Point Identifier is disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:HOP:TCAP:TCTim ON</p> <p>SENS:DATA:TEL:SDHS:HOP:TCAP:TCTim?</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM?

---



## Streams - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:MODE**

---

<b>Description</b>	<p>This command sets the Transmit Mode.</p> <p>At *RST condition, this value is set to Continuous.</p> <p>Navigation Path: EtherBERT &gt; Test Setup &gt; Test Configurator &gt; Shaping &gt; Transmit Mode</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:MODE &lt;wsp&gt;CONTInuous   NFRame</p>
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CONTInuous   NFRame</p> <p>Sets the transmit mode for the stream configuration.</p> <p>CONTInuous: Continuous</p> <p>NFRame: Number of frames</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TRAN:MODE CONTINUOUS</p> <p>SOUR:DATA:TEL:ETH:STR:TRAN:MODE?</p> <p>Returns: CONTINUOUS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:MODE?</p>

---

## SCPI Command Reference

### Streams - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:MODE?**

<b>Description</b>	<p>This query returns the Transmit Mode.</p> <p>At *RST condition, this value is set to Continuous.</p> <p>Navigation Path: EtherBERT &gt; Test Setup &gt; Test Configurator &gt; Shaping &gt; Transmit Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the transmit mode of the stream configuration.</p> <p>CONTinuous, selects Continuous as the transmit mode.</p> <p>NFRame, selects Number of frames as the transmit mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:TRAN:MODE CONTINUOUS</p> <p>SOUR:DATA:TEL:ETH:STR:TRAN:MODE?</p> <p>Returns: CONTINUOUS</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:MODE?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame**

---

<b>Description</b>	<p>This command sets the number of frames.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: EtherBERT &gt; Test Setup &gt; Test Configurator &gt; Shaping &gt; Frame Count</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame &lt;wsp&gt; &lt;Size&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Size:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the number of frames.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:TRAN:NFR 100 SOUR:DATA:TEL:ETH:STR:TRAN:NFR? Returns: 100</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame?</code>

---

## SCPI Command Reference

### Streams - Profile

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame?

---

<b>Description</b>	This query returns the number of frames. At *RST condition, this value is set to 1. Navigation Path: EtherBERT > Test Setup > Test Configurator > Shaping > Frame Count
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame?
<b>Response Syntax</b>	<Frame>
<b>Response(s)</b>	<b>Frame:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the number of frames for each traffic type. MAXimum, selects maximum as number of frames. MINimum, selects minimum as number of frames.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:TRAN:NFR 100 SOUR:DATA:TEL:ETH:STR:TRAN:NFR? Returns: 100
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:TRANsmit:NFRame

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE:TYPE**

<b>Description</b>	<p>This command sets the frame size type.</p> <p>At *RST condition, this value is set to Fixed.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Ethernet Frame &gt; Frame Size &gt; Selection (dropdown)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE:TYPE <wsp> <Stream>, FIXED   RANDOM   EMIX   SWEEP
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FIXED   RANDOM   EMIX   SWEEP</p> <p>Sets the frame size type for selected stream.</p> <p>FIXED</p> <p>RANDOM</p> <p>EMIX</p> <p>SWEEP</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:FRAM:SIZE:TYPE 1, FIXED</p> <p>SOUR:DATA:TEL:ETH:STR:FRAM:SIZE:TYPE? 1</p> <p>Returns: FIXED</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEEp:START?

## SCPI Command Reference

### Streams - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE:TYPE?**

<b>Description</b>	<p>This query returns the frame size type.</p> <p>At *RST condition, this value is set to Fixed.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Ethernet Frame &gt; Frame Size &gt; Selection (dropdown)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE:TYPE? &lt;wsp&gt; &lt;Stream&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Frame Size Type&gt;</p>
<b>Response(s)</b>	<p><b>Frame Size Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type frame size for the selected stream.</p> <p>FIXED, returns the frame type as FIXED.</p> <p>RANDOM, returns the frame type as RANDOM.</p> <p>EMIX, returns the frame type as EMIX.</p> <p>SWEEP, returns the frame type as SWEEP.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:FRAM:SIZE:TYPE 1, FIXED</p> <p>SOUR:DATA:TEL:ETH:STR:FRAM:SIZE:TYPE? 1</p> <p>Returns: FIXED</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEEp:STARt</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE**

<b>Description</b>	<p>This command sets a unique frame size.</p> <p>At *RST condition, this value is set to 92 bytes.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Ethernet Frame &gt; Frame Size (text box)</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE &lt;wsp&gt; &lt;Stream&gt;, &lt;Size&gt;</code>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets a unique frame size. Selects the stream from 1 to 10.</p> <p><b>Size:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the frame size for selected stream.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:STR:FRAM:SIZE:TYPE 1, FIXED</code> <code>SOUR:DATA:TEL:ETH:STR:FRAM:SIZE 1, 500</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE:TYPE 1, FIXED</code>

---

## SCPI Command Reference

### Streams - Profile

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE?

---

<b>Description</b>	<p>This query returns the frame size.</p> <p>At *RST condition, this value is set to 92 bytes.</p> <p>Navigation Path: EthernetBERT &gt; Test Setup &gt; Test Configurator &gt; Ethernet Frame &gt; Frame Size (text box)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE? <wsp> <Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 10.</p>
<b>Response Syntax</b>	<Frame size>
<b>Response(s)</b>	<p><b>Frame size:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frame size for the stream.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:FRAM:SIZE:TYPE 1, FIXED</p> <p>SOUR:DATA:TEL:ETH:STR:FRAM:SIZE 1, 500</p> <p>SOUR:DATA:TEL:ETH:STR:FRAM:SIZE? 1</p> <p>Returns: 500</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE:TYPE? 1

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE**

<b>Description</b>	<p>This command sets the transmitter mode for the selected stream for Data profile type.</p> <p>At *RST condition, this value is set to CONTInuous.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt; TX Mode</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE &lt;wsp&gt; &lt;Tgen&gt;, CONTInuous   BURSt   RAMP   NFRame   NBURst   NRAMp</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Mode:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CONTInuous   BURSt   RAMP   NFRame   NBURst   NRAMp</p> <p>Selects the transmitter mode for the selected stream for Data profile type.</p> <p>CONTInuous: Continuous.</p> <p>BURSt: Burst.</p> <p>RAMP: Ramp.</p> <p>NFRame: Number of Frame.</p> <p>NBURst: Number of Burst.</p> <p>NRAMp: Number of Ramp.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, BURS</p> <p>SOUR:DATA:TEL:ETH:STR:MODE? 1</p> <p>Returns: BURST</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VIDeo:CHANnels</p>

## SCPI Command Reference

### Streams - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE?**

<b>Description</b>	<p>This query returns the transmitter mode for stream 1.</p> <p>At *RST condition, this value is set to CONTInuous.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt; TX Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the transmitter mode for Stream 1.</p> <p>CONTINUOUS, Continuous is selected as transmitter mode.</p> <p>BURST, Burst is selected as transmitter mode.</p> <p>RAMP, Ramp is selected as transmitter mode.</p> <p>NFRAME, Non Frame is selected as transmitter mode.</p> <p>NBURST, Non Burst is selected as transmitter mode.</p> <p>NRAMP, Non Ramp is selected as transmitter mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, BURS</p> <p>SOUR:DATA:TEL:ETH:STR:MODE? 1</p> <p>Returns: BURST</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VIDeo:CHANnels?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:COUNT**

<b>Description</b>	<p>This command sets the ramp cycle count for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode nRamp&gt; Ramp Cycle Count</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:COUNT &lt;wsp&gt; &lt;Tgen&gt;, &lt;Count&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Count:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the ramp cycle count.</p> <p>Choices are 1 through 225.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, RAMP</p> <p>SOUR:DATA:TEL:ETH:STR:RAMP:COUN 1, 20</p> <p>SOUR:DATA:TEL:ETH:STR:RAMP:COUN? 1</p> <p>Returns: 20</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNT?</p>

---

## SCPI Command Reference

### Streams - Profile

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:COUNT?

---

<b>Description</b>	<p>This query returns the number of ramp cycle count for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode nRamp&gt; Ramp Cycle Count</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:COUNT? <wsp> <Tgen>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Select the number of ramp.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the ramp count value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, RAMP</p> <p>SOUR:DATA:TEL:ETH:STR:RAMP:COUN 1, 20</p> <p>SOUR:DATA:TEL:ETH:STR:RAMP:COUN? 1</p> <p>Returns: 20</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:STEP**

<b>Description</b>	<p>This command sets the number of ramp steps for the selected traffic stream.</p> <p>At *RST condition, this value is set to 10.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode Ramp/nRamp&gt; Ramp Number Of Steps</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:STEP <wsp> <Tgen>, <Step>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Step:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Set the step for the ramp.</p> <p>Choices are 2 through 100.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, RAMP</p> <p>SOUR:DATA:TEL:ETH:STR:RAMP:STEP 1, 50</p> <p>SOUR:DATA:TEL:ETH:STR:RAMP:STEP? 1</p> <p>Returns: 50</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE

## SCPI Command Reference

### Streams - Profile

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:STEP?

<b>Description</b>	<p>This query returns the number of ramp steps for the selected traffic stream.</p> <p>At *RST condition, this value is set to 10.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode Ramp/nRamp&gt; Ramp Number Of Steps</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:STEP? <wsp> <Tgen>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Select the step for the ramp.</p>
<b>Response Syntax</b>	<Step>
<b>Response(s)</b>	<p><b>Step:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the ramp step value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, RAMP</p> <p>SOUR:DATA:TEL:ETH:STR:RAMP:STEP 1, 50</p> <p>SOUR:DATA:TEL:ETH:STR:RAMP:STEP? 1</p> <p>Returns: 50</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:TIME**

---

<b>Description</b>	<p>This command sets the ramp step time for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1000.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode Ramp/nRamp&gt; Ramp Time</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:TIME &lt;wsp&gt; &lt;Tgen&gt;, &lt;Time&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Time:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Set the ramp step time.</p> <p>Choices are 100 through 8000 milliseconds.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:MODE 1, RAMP SOUR:DATA:TEL:ETH:STR:RAMP:TIME 1, 5 SOUR:DATA:TEL:ETH:STR:RAMP:TIME? 1 Returns: 5</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:TIME</code>

---

## SCPI Command Reference

### Streams - Profile

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:TIME?

---

<b>Description</b>	<p>This query returns the ramp step time for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1000.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode Ramp/nRamp&gt; Ramp Time</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:TIME? <wsp> <Tgen>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Select the ramp time.</p>
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the ramp step time.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, RAMP</p> <p>SOUR:DATA:TEL:ETH:STR:RAMP:TIME 1, 5</p> <p>SOUR:DATA:TEL:ETH:STR:RAMP:TIME? 1</p> <p>Returns: 5</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:TIME?

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled**

<b>Description</b>	<p>This command enables or disables the status of the stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Enable TX</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Global &gt; Enable Stream (Check box in table before Stream Name)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled <wsp> <Tgen>, ON   OFF
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Set the status of transmitter.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:ENAB 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:ENAB? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:STATus

## SCPI Command Reference

### Streams - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled?**

<b>Description</b>	<p>This query returns the status of the stream.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Enable TX</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Global &gt; Enable Stream (Check box in table before Stream Name)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ENABled? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:ENAB 1, ON</p> <p>SOUR:DATA:TEL:ETH:STR:ENAB? 1</p> <p>Returns: 2</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:STATus?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:TOTal:BANDwidth?**

---

<b>Description</b>	<p>This query returns the total available bandwidth value for all configured traffic streams.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Link Capacity</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:TOTal:BANDwidth?
<b>Response Syntax</b>	<Bandwidth>
<b>Response(s)</b>	<p><b>Bandwidth:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the total available bandwidth.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:TOT:BAND?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ENABLEd:BANDwidth?

---

## SCPI Command Reference

### Streams - Profile

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ENABled:BANDwidth?

---

<b>Description</b>	<p>This query returns the total enabled bandwidth value for all the configured traffic streams.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Total Tx Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ENABled:BANDwidth?
<b>Response Syntax</b>	<Bandwidth>
<b>Response(s)</b>	<p><b>Bandwidth:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Return the total enabled bandwidth.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ENAB:BAND?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:TOTal:BANDwidth?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FCOut**

---

<b>Description</b>	<p>This command counts the number of frames transmitted for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt; Frame Count</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FCOut &lt;wsp&gt; &lt;Tgen&gt;, &lt;Fcount&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Fcount:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects number of frame counts.</p>
<b>Example(s)</b>	<pre>SOURce:DATA:TELEcom:ETHernet:STReam:MODE 1, NFR SOUR:DATA:TEL:ETH:STR:FCO 1, 500 SOUR:DATA:TEL:ETH:STR:FCO? 1 Returns: 500</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE</code>

---

## SCPI Command Reference

### Streams - Profile

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FCOut?

---

<b>Description</b>	<p>This query returns the number of frames transmitted for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt; Frame Count</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Global &gt; TX Frame Count</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FCOut? <wsp> <Tgen>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets number of counts.</p>
<b>Response Syntax</b>	<Fcount>
<b>Response(s)</b>	<p><b>Fcount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of frame counts.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, NFR</p> <p>SOUR:DATA:TEL:ETH:STR:FCO 1, 500</p> <p>SOUR:DATA:TEL:ETH:STR:FCO? 1</p> <p>Returns: 500</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE?

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:RATE**

<b>Description</b>	<p>This command sets the stream rate for the selected traffic stream.</p> <p>At *RST condition, this value is set to 100.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt; TX Rate</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:RATE &lt;wsp&gt; &lt;Tgen&gt;, &lt;Rate&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Rate:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the stream rate for the transmitter.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:PROFile:RATE 1, 105 SOUR:DATA:TEL:ETH:STR:PROFile:RATE? 1 Returns: 105</pre>
<b>See Also</b>	<code>SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:FCOunt</code>

## SCPI Command Reference

### Streams - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:RATE?**

<b>Description</b>	<p>This query returns stream rate for the selected traffic stream.</p> <p>At *RST condition, this value is set to 100.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt; TX Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:RATE? &lt;wsp&gt; &lt;Tgen&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Select the rate for the transmitter.</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the stream rate for the transmitter.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:PROFile:RATE 1, 105</p> <p>SOUR:DATA:TEL:ETH:STR:PROFile:RATE? 1</p> <p>Returns: 105</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FCOunt?</p>



<b>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:STARt</b>	
<b>Description</b>	<p>This command sets the sweep frame start size.</p> <p>At *RST condition, this value is set to 64.</p> <p>Navigation Path: Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams&gt;Profile&gt;Frame Size(Bytes)&gt;Sweep&gt;Sweep Range(1 byte step)&gt; Min(Bytes)</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:STARt &lt;wsp&gt; &lt;Stream&gt;, &lt;Sweep Range&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Sweep Range:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the sweep frame start size.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:STR:FRAM:SWE:STAR 1,64</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE?</code>

## SCPI Command Reference

### Streams - Profile

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:STARt?

---

<b>Description</b>	<p>This query returns the sweep frame start size.</p> <p>At *RST condition, this value is set to 64.</p> <p>Navigation Path: Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams&gt;Profile&gt;Frame Size(Bytes)&gt;Sweep&gt;Sweep Range(1 byte step)&gt; Min(Bytes)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:STARt? <wsp> <Stream>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current sweep frame start size is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Frame Size Type>
<b>Response(s)</b>	<p><b>Frame Size Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the sweep frame start size.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:FRAM:SWE:STAR 1,64</p> <p>SOUR:DATA:TEL:ETH:STR:FRAM:SWE:STAR? 1</p> <p>Returns: 64</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SIZE

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:END**

<b>Description</b>	<p>This command sets the sweep frame end size.</p> <p>At *RST condition, this value is set to 1518.</p> <p>Navigation Path: Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams&gt;Profile&gt;Frame Size(Bytes)&gt;Sweep&gt;Sweep Range(1 byte step)&gt; Max(Bytes)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:END &lt;wsp&gt; &lt;Stream&gt;, &lt;Sweep Range&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Sweep Range:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the sweep frame end size.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:FRAM:SWE:END 1,67</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP?</p>

## SCPI Command Reference

### Streams - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:END?**

<b>Description</b>	<p>This query returns the sweep frame end size.</p> <p>At *RST condition, this value is set to 1518.</p> <p>Navigation Path: Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams&gt;Profile&gt;Frame Size(Bytes)&gt;Sweep&gt;Sweep Range(1 byte step)&gt; Max(Bytes)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:FRAMe:SWEep:END? &lt;wsp&gt; &lt;Stream&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current sweep frame end size is returned.</p> <p>MAXimum: Biggest supported value MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Frame Size Type&gt;</p>
<b>Response(s)</b>	<p><b>Frame Size Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the sweep frame end size.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:FRAM:SWE:END 1,67 SOUR:DATA:TEL:ETH:STR:FRAM:SWE:END? 1 Returns: 67</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDReSS:DESTination:IP</p>

## Streams - Profile (Profile)

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE**

---

<b>Description</b>	<p>This command sets the profile type for the selected traffic stream.</p> <p>At *RST condition, this value is set to Data.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE <wsp> <Tgen>, VOICe   VIDEo   DATA
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Profile:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: VOICe   VIDEo   DATA</p> <p>Sets the profile type.</p> <p>VOICe: Voice</p> <p>VIDeo: Video</p> <p>DATA: Data</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:PROF:TYPE 2,VOICE</p> <p>SOUR:DATA:TEL:ETH:STR:PROF:TYPE? 2</p> <p>Returns: VOICE</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VOICE

---

## SCPI Command Reference

### *Streams - Profile (Profile)*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE?**

<b>Description</b>	<p>This query returns the profile type for the selected traffic stream.</p> <p>At *RST condition, this value is set to Data.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Profile>
<b>Response(s)</b>	<p><b>Profile:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the profile type.</p> <p>VOICe, Voice as profile type is selected.</p> <p>VIDeo, Video as profile type is selected.</p> <p>DATA, Data as profile type is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:PROF:TYPE 2,VOIC</p> <p>SOUR:DATA:TEL:ETH:STR:PROF:TYPE? 2</p> <p>Returns: VOICE</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODEc:VOICe?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VOICe**

<b>Description</b>	<p>This command sets the voice codec for the selected traffic stream.</p> <p>At *RST condition, this value is set to VG711.</p> <p>Navigation Path: Traffic Gen &amp; Mon &gt; Test Setup &gt; Test Configurator &gt; Stream &gt; Profile &gt; Voice &gt; Voice Codec</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VOICe &lt;wsp&gt; &lt;Tgen&gt;, VG711   VG723   VG729</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Voice:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: VG711   VG723   VG729</p> <p>Sets the voice codec for the selected traffic stream..</p> <p>VG711: VG711</p> <p>VG723: VG723</p> <p>VG729: VG729</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:COD:VOICe 2, VG723</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VOICe? 2</p> <p>Returns: VG723</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE</p>

## SCPI Command Reference

### *Streams - Profile (Profile)*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VOICe?**

<b>Description</b>	<p>This query returns the voice codec for the selected traffic stream.</p> <p>At *RST condition, this value is set to VG711.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Voice &gt;Voice Codec</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VOICe? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Jitter>
<b>Response(s)</b>	<p><b>Jitter:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the voice codec selected.</p> <p>VG711, VG711 as jitter packet type is selected.</p> <p>VG723, VG723 as jitter packet type is selected.</p> <p>VG729, VG729 as jitter packet type is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:COD:VOICe 2, VG723</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VOICe? 2</p> <p>Returns: VG723</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:TYPE?



**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VOICe:CALLs**

<b>Description</b>	<p>This command sets the voice call codec for the selected traffic stream.</p> <p>At *RST condition, this value is set to 10.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Voice &gt;Number Of Calls</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VOICe:CALLs <wsp> <Tgen>, <Calls>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Calls:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the number of calls.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:COD:VOICe 2,VG723</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VOIC:CALL 2,20</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VOIC:CALL? 2</p> <p>Returns: 20</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VIDeo:CHANnels?

## SCPI Command Reference

### *Streams - Profile (Profile)*

---

<b>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VOICe:CALLs?</b>	
<b>Description</b>	<p>This query returns the number of calls for the selected traffic stream, when Voice is selected.</p> <p>At *RST condition, this value is set to 10.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Voice &gt;Number Of Calls</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VOICe:CALLs? &lt;wsp&gt; &lt;Tgen&gt;[, MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p>
<b>Response Syntax</b>	<code>&lt;Calls&gt;</code>
<b>Response(s)</b>	<p><b>Calls:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of calls.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:STR:COD:VOICe 2,VG723</code> <code>SOUR:DATA:TEL:ETH:STR:COD:VOIC:CALL 2,20</code> <code>SOUR:DATA:TEL:ETH:STR:COD:VOIC:CALL? 2</code> Returns: 20
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VOICe:CHANnels?</code>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VIDeo**

---

<b>Description</b>	<p>This command sets the video codec for the selected traffic stream.</p> <p>At *RST condition, this value is set to SDTVMPEG2.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Video &gt;Video Codec</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VIDeo &lt;wsp&gt; &lt;Tgen&gt;, SDTVMPEG2   HDTV MPEG2   HDTV MPEG4</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Profile:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SDTVMPEG2   HDTV MPEG2   HDTV MPEG4</p> <p>Sets the video coded type packet type.</p> <p>SDTVMPEG2, sets SDTV (MPEG-2).</p> <p>HDTV MPEG2, sets HDTV (MPEG-2).</p> <p>HDTV MPEG4, sets HDTV (MPEG-4).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:PROF:TYPE 2,VIDeo</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VID 2,SDTVMPEG2</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VID? 2</p> <p>Returns: SDTVMPEG2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODeC:VIDeo?</p>

---

## SCPI Command Reference

### *Streams - Profile (Profile)*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VIDeo?**

<b>Description</b>	<p>This query returns the video codec for the selected traffic stream.</p> <p>At *RST condition, this value is set to SDTVMPEG2.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Video &gt;Video Codec</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VIDeo? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Codec>
<b>Response(s)</b>	<p><b>Codec:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the video codec type for the selected stream.</p> <p>SDTVMPEG2, SDTV (MPEG-2) is selected as video codec type.</p> <p>HDTVMPEG2, HDTV (MPEG-2) is selected as video codec type.</p> <p>HDTVMPEG4, HDTV (MPEG-4) is selected as video codec type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:PROF:TYPE 2,VIDeo</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VID 2,SDTVMPEG2</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VID? 2</p> <p>Returns: SDTVMPEG2</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VIDeo:CHANnels**

<b>Description</b>	<p>This command sets the video channel for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Video &gt;Number Of Channels</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VIDeo:CHANnels <wsp> <Tgen>, <Channels>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Channels:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the number of channels.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:PROF:TYPE 2,VIDeo</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VID 2,SDTVMPEG2</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VID:CHAN 2,20</p> <p>SOUR:DATA:TEL:ETH:STR:COD:VID:CHAN? 2</p> <p>Returns: 20</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE?

## SCPI Command Reference

### *Streams - Profile (Profile)*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VIDeo:CHANnels?**

<b>Description</b>	<p>This query returns the video channel for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Video &gt;Number Of Channels</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VIDeo:CHANnels? &lt;wsp&gt; &lt;Tgen&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p>
<b>Response Syntax</b>	<p>&lt;Channels&gt;</p>
<b>Response(s)</b>	<p><b>Channels:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of channels.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:PROF:TYPE 2,VIDeo SOUR:DATA:TEL:ETH:STR:COD:VID 2,SDTVMPEG2 SOUR:DATA:TEL:ETH:STR:COD:VID:CHAN 2,20 SOUR:DATA:TEL:ETH:STR:COD:VID:CHAN? 2 Returns: 20</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:CODec:VIDeo</p>

# Shaping

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:BANDwidth**

---

<b>Description</b>	<p>This command sets the burst bandwidth for the selected traffic stream.</p> <p>At *RST condition, this value is set to 50.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode Burst /nBurst &gt; Burst Duty Cycle%</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:BANDwidth &lt;wsp&gt; &lt;Tgen&gt;, &lt;Bandwidth&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Bandwidth:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Set the bandwidth in percentage.</p> <p>Choices are 1 through 100%.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, BURS</p> <p>SOUR:DATA:TEL:ETH:STR:BURS:BAND 1, 100</p> <p>SOUR:DATA:TEL:ETH:STR:BURS:BAND? 1</p> <p>Returns: 100</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE</p>

---

## SCPI Command Reference

### Shaping

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:BANDwidth?**

<b>Description</b>	<p>This query returns the burst bandwidth for the selected traffic stream.</p> <p>At *RST condition, this value is set to 50.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode Burst /nBurst &gt; Burst Duty Cycle %</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:BANDwidth? &lt;wsp&gt; &lt;Tgen&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Select the bandwidth for the transmitter.</p>
<b>Response Syntax</b>	<p>&lt;Bandwidth&gt;</p>
<b>Response(s)</b>	<p><b>Bandwidth:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the burst bandwidth from 1 to 100 in percentage.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, BURS</p> <p>SOUR:DATA:TEL:ETH:STR:BURS:BAND 1, 100</p> <p>SOUR:DATA:TEL:ETH:STR:BURS:BAND? 1</p> <p>Returns: 100</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MODE?</p>



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNT**

---

<b>Description</b>	<p>This command sets the number of burst count for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode nBurst &gt; Burst Count</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNT &lt;wsp&gt; &lt;Tgen&gt;, &lt;Count&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Count:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Set the number of the burst count.</p> <p>Choices are 1 through 255.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:STR:MODE 1, BURS SOUR:DATA:TEL:ETH:STR:BURS:COUN 1, 50 SOUR:DATA:TEL:ETH:STR:BURS:COUN? 1 Returns: 50</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:BANDwidth</code>

---

## SCPI Command Reference

### Shaping

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNT?

<b>Description</b>	<p>This query returns the number of burst count for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode nBurst &gt; Burst Count</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNT? <wsp> <Tgen>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Select the number of the burst.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the burst count.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, BURS</p> <p>SOUR:DATA:TEL:ETH:STR:BURS:COUN 1, 50</p> <p>SOUR:DATA:TEL:ETH:STR:BURS:COUN? 1</p> <p>Returns: 50</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:BANDwidth?

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:TIME**

<b>Description</b>	<p>This command sets the burst time for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1000.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode Burst /nBurst &gt; Period</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:TIME <wsp> <Tgen>, <Time>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Time:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the burst time.</p> <p>Choices are 1 to 8000 milliseconds.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, BURS</p> <p>SOUR:DATA:TEL:ETH:STR:BURS:TIME 1, 50</p> <p>SOUR:DATA:TEL:ETH:STR:BURS:TIME? 1</p> <p>Returns: 50</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNT

## SCPI Command Reference

### Shaping

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:TIME?**

<b>Description</b>	<p>This query returns the duration of burst for the selected traffic stream.</p> <p>At *RST condition, this value is set to 1000.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Shaping &gt;Tx Mode Burst /nBurst &gt; Period</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:TIME? &lt;wsp&gt; &lt;Tgen&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Select the duration.</p>
<b>Response Syntax</b>	<p>&lt;Time&gt;</p>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns burst time.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:MODE 1, BURS</p> <p>SOUR:DATA:TEL:ETH:STR:BURS:TIME 1, 50</p> <p>SOUR:DATA:TEL:ETH:STR:BURS:TIME? 1</p> <p>Returns: 50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BURSt:COUNT?</p>

## Streams - Global

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:REStore:DEFault**

---

<b>Description</b>	<p>This command restores the stream configuration to default values.</p> <p>This is action command so no default values.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Streams &gt;Global &gt; Restore Default</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:REStore:DEFault
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:STReam:GLOBal:REStore:DEFault
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RAMP:STEP?

---

## SCPI Command Reference

### *Streams - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABle**

<b>Description</b>	<p>This command sets the QOS Metrics Tag Insertion enable for Traffic gen test.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Global &gt; QOS Metrics tag insertion</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABle
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables QOS Metrics Tag Insertion.</p>
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABle ON
<b>See Also</b>	SOURce[1..n]:DATA:TEL:ETH:STR:PROFile:RATE

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABLE?**

<b>Description</b>	<p>This query return the QOS Metrics Tag Insertion enable for Traffic gen test.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Global &gt; QOS Metrics tag insertion</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABLE?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of QOS metrics tag insertion.</p>
<b>Example(s)</b>	<p>SOURCE:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABLE ON</p> <p>SOURCE:DATA:TELEcom:ETHernet:STReam:GLOBal:QOSMetrics:ENABLE?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TEL:ETH:STR:PROFile:RATE?

---

## SCPI Command Reference

### *Streams - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NAME**

<b>Description</b>	<p>This command sets the stream name for selected traffic stream.</p> <p>At *RST condition, this value is set to iStream1 for first stream.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Stream Name</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Streams &gt;Global &gt; Stream Name</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NAME &lt;wsp&gt; &lt;Tgen&gt;, &lt;Name&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Name:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the name for selected stream.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:STReam:NAME 1, streamA</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:RATE?</p>

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NAME?**

<b>Description</b>	<p>This query returns stream name for the selected traffic stream.</p> <p>At *RST condition, this value is set to iStreamI1 for first stream.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Profile &gt; Stream Name</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Streams &gt;Global &gt; Stream Name</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:NAME? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Name>
<b>Response(s)</b>	<p><b>Name:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the name for selected stream</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:STReam:NAME 1, streamA</p> <p>SOURce:DATA:TELEcom:ETHernet:STReam:NAME? 1</p> <p>Returns: streamA</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:PROFile:RATE

## SCPI Command Reference

### *Streams - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABle**

<b>Description</b>	<p>This command enables/disables the Source IP Address group for batch Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABle &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable or disable the Source IP Address group.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:IP:ENAB ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP:ENABle</p>

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABle?**

<b>Description</b>	<p>This query returns the status of the Source IP Address group for batch Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABle?
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<p><b>set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Source IP Address group status.</p> <p>1, Source IP Address group is enabled.</p> <p>0, Source IP Address group is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:IP:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:IP:ENAB?</p> <p>Return: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP:ENABle?

---

## SCPI Command Reference

### *Streams - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDRess:TYPE**

<b>Description</b>	<p>This command sets the Source IP Address type for batch Configuration.</p> <p>At *RST condition, this value is set to ""Couple with Interface"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address &gt; Couple With Interface / Automatic IP (DHCP) / Set to</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDRess:TYPE &lt;wsp&gt;COUPLEWITHINTERFACE   DHCP   STATICIP</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: COUPLEWITHINTERFACE   DHCP   STATICIP</p> <p>Selects Source IP Address type.</p> <p>COUPLEWITHINTERFACE: Couple With Interface</p> <p>DHCP: Automatic IP (DHCP)</p> <p>STATICIP: Static IP</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:ADDR:TYPE DHCP</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:TYPE</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDRess:TYPE?**

---

<b>Description</b>	<p>This query returns the Source IP Address type for batch Configuration.</p> <p>At *RST condition, this value is set to ""Couple with Interface"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address &gt; Couple With Interface / Automatic IP (DHCP) / Set to</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDRess:TYPE?
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<p><b>set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Source IP Address type.</p> <p>COUPLEWITHINTERFACE, Couple With Interface as Source IP Address type.</p> <p>DHCP, Automatic IP (DHCP) as Source IP Address type.</p> <p>STATICIP, Set to as Source IP Address type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:ADDR:TYPE DHCP</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:ADDR:TYPE?</p> <p>Returns:: DHCP</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:TYPE?

---

## SCPI Command Reference

### *Streams - Global*

---

---

#### **:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURCE:IP**

---

<b>Description</b>	<p>This command sets the Source IP Address for batch Configuration.</p> <p>At *RST condition, this value is set to ""10.10.0.0"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM&gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address &gt; IP address</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURCE:IP &lt;wsp&gt; &lt;address&gt;</code>
<b>Parameter(s)</b>	<p><b>address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Source IP Address.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:IP "10.10.10.10"</code>
<b>See Also</b>	<code>SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP</code>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP?**

---

<b>Description</b>	<p>This query returns the Source IP Address for batch Configuration.</p> <p>At *RST condition, this value is set to ""10.10.0.0"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address &gt; IP address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP?
<b>Response Syntax</b>	<address>
<b>Response(s)</b>	<p><b>address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Source IP address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:IP "10.10.10.10"</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:IP?</p> <p>Returns: "10.10.10.10"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP?

---

## SCPI Command Reference

### *Streams - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:SUBNet:MASK**

<b>Description</b>	<p>This command sets the Subnet mask Address for batch Configuration.</p> <p>At *RST condition, this value is set to ""255.255.0.0"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address &gt; Subnet mask</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:SUBNet:MASK &lt;wsp&gt; &lt;address&gt;</p>
<b>Parameter(s)</b>	<p><b>address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Subnet mask Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:SUBN:MASK "255.255.255.0"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:IP</p>

---



---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURCE:SUBNet:MASK?**

---

<b>Description</b>	<p>This query returns the Subnet mask Address for batch Configuration.</p> <p>At *RST condition, this value is set to ""255.255.0.0"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address &gt; Subnet mask</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURCE:SUBNet:MASK?
<b>Response Syntax</b>	<address>
<b>Response(s)</b>	<p><b>address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Subnet mask address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:SUBN:MASK "255.255.255.0"</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:SOUR:SUBN:MASK?</p> <p>Returns: "255.255.255.0"</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFAult:GATeway:IP?

---

## SCPI Command Reference

### *Streams - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:ENABle**

<b>Description</b>	<p>This command enables/disables the Default gateway for batch Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address &gt; Default gateway</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:ENABle &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable or disable the Default gateway.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEF:GAT:ENAB ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABle</p>

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:ENABle?**

<b>Description</b>	<p>This query returns the status of the Default gateway for batch Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address &gt; Default gateway</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:ENABle?
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<p><b>set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Default gateway status.</p> <p>1, Default gateway is enabled.</p> <p>0, Default gateway is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEF:GAT:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:DEF:GAT:ENAB?</p> <p>Return: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABle?

---

## SCPI Command Reference

### *Streams - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:IP**

<b>Description</b>	<p>This command sets the Default gateway IP Address for batch Configuration.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address &gt; Default gateway</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:IP &lt;wsp&gt; &lt;address&gt;</p>
<b>Parameter(s)</b>	<p><b>address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Default gateway IP Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEF:GAT:IP "10.10.10.10"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP</p>

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:IP?**

<b>Description</b>	<p>This query returns the Default gateway IP Address for batch Configuration.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Source IP address &gt; Default gateway</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:IP?
<b>Response Syntax</b>	<address>
<b>Response(s)</b>	<p><b>address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Default gateway IP address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEF:GAT:IP "10.10.10.10"</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:DEF:GAT:IP?</p> <p>Returns: "10.10.10.10"</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP?

## SCPI Command Reference

### *Streams - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP:ENABle**

<b>Description</b>	<p>This command enables/disables the Destination IP Address group for batch Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Destination IP address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP:ENABle &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable or disable the Destination IP Address group.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:IP:ENAB ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ENABle</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP:ENABle?**

<b>Description</b>	<p>This query returns the status of the Destination IP Address group for batch Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Destination IP address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP:ENABle?
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<p><b>set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Destination IP Address group status.</p> <p>1, Destination IP Address group is enabled.</p> <p>0, Destination IP Address group is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:IP:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:IP:ENAB?</p> <p>Return: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ENABle?

---

## SCPI Command Reference

### *Streams - Global*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP**

---

<b>Description</b>	<p>This command sets the Destination IP Address for batch Configuration.</p> <p>At *RST condition, this value is set to ""10.10.0.0"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Destination IP address &gt; IP address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP &lt;wsp&gt; &lt;address&gt;</p>
<b>Parameter(s)</b>	<p><b>address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Destination IP Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:IP "10.10.10.10"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP</p>

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP?**

<b>Description</b>	<p>This query returns the Destination IP Address for batch Configuration.</p> <p>At *RST condition, this value is set to ""10.10.0.0"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Destination IP address &gt; IP address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:IP?
<b>Response Syntax</b>	<address>
<b>Response(s)</b>	<p><b>address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Destination IP address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:IP "10.10.10.10"</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:IP?</p> <p>Returns: "10.10.10.10"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP?

---

## SCPI Command Reference

### *Streams - Global*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ENABLE**

---

<b>Description</b>	<p>This command enables/disables the Destination MAC address group for batch Configuration. At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM&gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Destination MAC address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ENABLE &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable or disable the Destination MAC address group.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:MAC:ENAB ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABLE</p>

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ENABLE?**

<b>Description</b>	<p>This query returns the status of the Destination MAC address group for batch Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Destination MAC address</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ENABLE?
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<p><b>set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Destination MAC address group status.</p> <p>1, Destination MAC address group is enabled.</p> <p>0, Destination MAC address group is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:MAC:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:MAC:ENAB?</p> <p>Return: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:IP:ENABLE?

---

## SCPI Command Reference

### *Streams - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:TYPE**

<b>Description</b>	<p>This command sets the Destination MAC address type for batch Configuration.</p> <p>At *RST condition, this value is set to ""Resolve"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM&gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Destination MAC address &gt; Resolve / Set to</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:TYPE &lt;wsp&gt;RESOLVE   STATICMAC</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RESOLVE   STATICMAC</p> <p>Selects Destination MAC address type.</p> <p>RESOLVE: Resolve.</p> <p>STATICMAC: Static IP</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:MAC:TYPE RESOLVE</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDRess:TYPE</p>

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:TYPE?**

<b>Description</b>	<p>This query returns the Destination MAC address type for batch Configuration.</p> <p>At *RST condition, this value is set to ""Resolve"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Destination MAC address &gt; Resolve / Set to</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:TYPE?
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<p><b>set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Destination MAC address type.</p> <p>RESOLVE, Resolve as Destination MAC address type.</p> <p>STATICMAC, Set to as Destination MAC address type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:MAC:TYPE RESOLVE</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:MAC:TYPE?</p> <p>Returns:: RESOLVE</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:ADDRess:TYPE?

---

## SCPI Command Reference

### *Streams - Global*

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ADDRes**

---

<b>Description</b>	<p>This command sets the Destination MAC address for batch Configuration.</p> <p>At *RST condition, this value is set to ""00:00:00:00:00:00"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Destination MAC address &gt; MAC address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ADDRes &lt;wsp&gt; &lt;address&gt;</p>
<b>Parameter(s)</b>	<p><b>address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Destination MAC address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:MAC:ADDR "00:AA:DD:CC:11:00"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:SUBNet:MASK</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ADDRess?**

<b>Description</b>	<p>This query returns the Destination MAC address for batch Configuration.</p> <p>At *RST condition, this value is set to ""00:00:00:00:00:00"".</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM&gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Destination MAC address &gt; MAC address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DESTination:MAC:ADDRess? ?
<b>Response Syntax</b>	<address>
<b>Response(s)</b>	<p><b>address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Destination MAC address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:MAC:ADDR "00:AA:DD:CC:11:00"</p> <p>SOUR:DATA:TEL:ETH:STR:BATC:DEST:MAC:ADDR?</p> <p>Returns: "00:AA:DD:CC:11:00"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:SOURce:SUBNet:MASK?

---

## SCPI Command Reference

### *Streams - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy**

<b>Description</b>	<p>This command Select all or unselect all or invert all stream for batch Configuration.</p> <p>This is an event so no *RST condition.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM&gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Streams&gt; (Un)Select All/Invert</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy &lt;wsp&gt;SELECTALL   UNSELECTALL   INVERT</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SELECTALL   UNSELECTALL   INVERT</p> <p>Sets the Streams.</p> <p>SELECTALL: all the streams for copy</p> <p>UNSELECTALL,, Deselect all the streams for copy</p> <p>INVERT, Invert the streams for copy</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATC:COpy SELECTALL</p> <p>Selects all the streams for copying</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:APPLy</p>



---

<b>:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:STReam</b>	
<b>Description</b>	<p>This command enable/disable the stream for batch Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Streams</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COpy:STReam <wsp><Stream>, ON   OFF
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the stream.</p> <p><b>set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable or disable the stream.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:BATCh:COpy:STR 1,ON
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFault:GATeway:IP

---

## SCPI Command Reference

### *Streams - Global*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPI:STReam?**

---

<b>Description</b>	<p>This query returns the enabled streams for batch Configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Streams</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPI:STReam?
<b>Response Syntax</b>	<result>
<b>Response(s)</b>	<p><b>result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the all selected streams</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:STR:BATCh:COPI:STR 1,ON</p> <p>SOUR:DATA:TEL:ETH:STR:BATCh:COPI:STR?</p> <p>Returns: "Enabled Stream = 1"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:DEFAult:GATeway:IP?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPIY:APPLy**

<b>Description</b>	<p>This command apply trigger copy to all selected stream for batch Configuration.</p> <p>This is an event so no *RST condition.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Apply to</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPIY:APPLy
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:STR:BATC:COPIY:APPL
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPIY

---

## SCPI Command Reference

### *Streams - Global*

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPI:SYNC:PROGress?**

---

<b>Description</b>	<p>This query returns the progress of copy triggered for batch Configuration.</p> <p>This is an event so no *RST condition.</p> <p>Navigation Path:Traffic Gen &amp; Mon/EtherSAM &gt; Test Setup &gt; Streams &gt; Global &gt; Addressing &gt; Batch &gt; Progress bar(applying changes...)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:BATCh:COPI:SYNC:PROGress?
<b>Response Syntax</b>	<Progress>
<b>Response(s)</b>	<p><b>Progress:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the progress of copy.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:STR:BATC:COPI:SYNC:PROG?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:CURREnt?

---

## Streams - Global (Copy Stream)

---

:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:COPIstream

---

<b>Description</b>	<p>This command copies the stream configuration of source stream to destination stream.</p> <p>There is no *RST value for this command.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Test Setup &gt; Test Configurator &gt; Stream &gt;Global &gt; Copy Stream</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:COPIstream <wsp> <Source Stream number>, <Destination stream>
<b>Parameter(s)</b>	<p><b>Source Stream number:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the source stream from 1 to 16.</p> <p><b>Destination stream:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the destination stream from 1 to 16.</p>
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:STReam:GLOBal:COPIstream 1, 2
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:REStore:DEFault

---

# Services - Profile

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:ENABle**

<b>Description</b>	<p>This command enables and disables the Traffic policing rate.</p> <p>At *RST condition, this value is set to Enabled.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt;Test Parameter&gt;Traffic Policing</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:ENABle &lt;wsp&gt; &lt;Service&gt;, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Enable:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the Traffic policing status as on or off.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:ENABle 1, ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:RESDefault</p>

---

---

```
:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:ENABle?
```

<b>Description</b>	<p>This query returns the Traffic policing status.</p> <p>At *RST condition, this value is set to Enabled.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt;Test Parameter&gt;Traffic Policing</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:ENABle? <wsp> <Service>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p>
<b>Response Syntax</b>	<status>
<b>Response(s)</b>	<p><b>status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Traffic policing status.</p> <p>1, Traffic policing is enabled.</p> <p>0, Traffic policing is disabled.</p>
<b>Example(s)</b>	<pre>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:ENABle 1, ON</pre> <pre>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:ENABle? 1</pre> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteriia: ENABle

---

## SCPI Command Reference

### Services - Profile

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:TRAPolicing:VALue**

<b>Description</b>	<p>This command sets the Traffic Policing Value for the selected service, direction, and parameter.</p> <p>At *RST Condition value is set to 100 (Mbits/s).</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt;Test Parameter&gt;Traffic Policing value</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARAmeter:TRAPolicing:VALue &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, &lt;Traffic Policing Value&gt;   MAXimum   MINimum</p>

---



---

```
:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:VALue
```

---

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Traffic Policing Value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets Traffic Policing Value for the selected Service.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:ENABLE 1, 1</pre> <pre>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:VALue 1, LTOR,99.0</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteri:a:ENABLE?</pre>

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:VALue?**

<b>Description</b>	<p>This command returns the Traffic Policing Value for the selected service, direction, and parameter.</p> <p>At *RST Condition value is set to 100 (Mbits/s).</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt;Test Parameter&gt;Traffic Policing value</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:VALue? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current Traffic Policing Value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:VALue?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Traffic policing value.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:ENABLE 1, 1 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:TRAPolicing:VALue? 1, LTOR
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:BMRate

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate:ENABLe**

<b>Description</b>	<p>This commands enables or disables the information rate for the selected service.</p> <p>At *RST condition CIR is enabled and CIR+EIR is disabled.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt; SLA Parameters Information Rate&gt;Information Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate:ENABLE &lt;wsp&gt; &lt;Service&gt;, CIR   CIREIR, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Information Rate Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CIR   CIREIR</p> <p>Selects the information rate for the selected service.</p> <p>CIR: CIR</p> <p>CIREIR: EIR</p> <p><b>Enable:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables the information rate for the selected service.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFR:ENABLe 1, Cir,1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAMeters:NOBSequence?</p>

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate:ENABle?**

<b>Description</b>	<p>This query returns the status of information rate for the selected service.</p> <p>At *RST condition CIR is enabled and CIR+EIR is disabled.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt; SLA Parameters Information Rate&gt;Information Rate</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate:ENABle? &lt;wsp&gt; &lt;Service&gt;, CIR   CIREIR</code>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Information Rate Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CIR   CIREIR</p> <p>Selects the information rate for the selected service.</p> <p>CIR: CIR</p> <p>CIREIR: EIR</p>
<b>Response Syntax</b>	<code>&lt;Status&gt;</code>

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate:ENABle?**

<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the information rate for the selected service. 1, information rate is enabled. 0, information rate is disabled.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFR:ENABle 1, Cir, 1 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFR:ENABle? 1, CIR Returns: 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:BURSt:PARAmeters:BIRFrame

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate**

---

<b>Description</b>	<p>This command sets the information rate for the selected service, direction, and parameter.</p> <p>At *RST condition, the rate is 50%* Line Rate for CIR and 75%*Line Rate for CIR+EIR.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt; SLA Parameters Information Rate&gt;Information Rate&gt;CIR,CIR+EIR</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate &lt;wsp&gt; &lt;Service&gt;, CIR   CIREIR, LTOR   RTOL, &lt;Value&gt;   MAXimum   MINimum</p>

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate**

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Information Rate Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CIR   CIREIR</p> <p>Selects the information rate for the selected service.</p> <p>CIR: CIR (Mbit/s)</p> <p>CIREIR: EIR +CIR (Mbit/s)</p> <p><b>Direction:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR, sets the direction from local to remote.</p> <p>RTOL, sets the direction from remote to local.</p> <p><b>Value:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value of the selected information rate type.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFR 1, CIR,LTOR,50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSequence?</p>



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate?**

---

<b>Description</b>	<p>This query returns the information rate for the selected service, direction, and parameter.</p> <p>At *RST condition, the rate is 50%* Line Rate for CIR and 75%*Line Rate for CIR+EIR.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt; SLA Parameters Information Rate&gt;Information Rate&gt;CIR,CIR+EIR</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate? &lt;wsp&gt; &lt;Service&gt;, CIR   CIREIR, LTOR   RTOL[, MAXimum   MINimum]</p>

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFIg:SERVIces:SLAParameter:INFRate?**

<b>Parameter(s)</b>	<b>Service:</b>
	<p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Information Rate Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CIR   CIREIR</p> <p>Selects the information rate for the selected service.</p> <p>CIR: CIR (Mbit/s)</p> <p>CIREIR: EIR +CIR (Mbit/s)</p> <p><b>Direction:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR, sets the direction from local to remote.</p> <p>RTOL, sets the direction from remote to local.</p> <p>The program data syntax for the fourth parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value of the selected information rate.</p> <p>This parameter is optional. If no token is specified, the current information rate value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFRate?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the configured value for the selected information rate type.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFR 1, CIR, LTOR, 50 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:INFR? 1, CIR, LTOR Returns: 50
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:BURSt:PARAmeters:RDERatio

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:EN  
ABLE**

**Description** This command enables performance criteria for the selected service, direction and parameter.  
At \*RST condition, the value is set to enabled for both direction.

Navigation Path: Test > Setup > Profile > SLA Parameters Performance Criteria

**Syntax** :SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PER  
Criteria:ENABle <wsp> <Service>, LTOR | RTOL, MAXJ | MAXRTL | MAXFL, ON | OFF

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:EN  
ABLE**

---

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Type:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXJ   MAXRTL   MAXFL</p> <p>Sets the performance criteria for the selected service.</p> <p>MAXJ: Max jitter (ms)</p> <p>MAXRTL: Max Round-Trip latency (ms)</p> <p>MAXFL: Max frame loss (%)</p> <p><b>Enable:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables performance criteria for the selected service.</p>
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:ENABLE 1, LTOR,MAXJ,1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAMeters:RDERatio?

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:EN  
ABLE?**

<b>Description</b>	<p>This query returns the status of performance criteria for the selected service, direction, and parameter.</p> <p>At *RST condition, the value is set to be enabled for both directions.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt; SLA Parameters Performance Criteria</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PER Criteria:ENABLE? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, MAXJ   MAXRTL   MAXFL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Type:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXJ   MAXRTL   MAXFL</p> <p>Sets the performance criteria for the selected service.</p> <p>MAXJ: Max jitter (ms)</p> <p>MAXRTL: Max Round-Trip latency (ms)</p> <p>MAXFL: Max frame loss (%)</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:EN  
ABLE?**

<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the performance criteria for the selected service. 1, performance criteria is enabled. 0, performance criteria is disabled.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:ENABle 1, LTOR,MAXJ,ON SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:ENABle? 1, LTOR,MAXJ Returns: 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VALue**

<b>Description</b>	<p>This command sets performance criteria values for the selected service, direction, and parameter.</p> <p>The *RST condition the values for Jitter to 2.0ms, Latency to 15.0ms and Frame Loss to 0.1%.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt; SLA Parameters Performance Criteria&gt;Max Jitter(ms)Round-Trip Latency(ms)Frame Loss (%)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VALue &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, MAXJ   MAXRTL   MAXFL, &lt;Value&gt;   MAXimum   MINimum</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPparameter:PERCriteria:VALue**

---

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Type:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXJ   MAXRTL   MAXFL</p> <p>Sets the performance criteria for the selected service.</p> <p>MAXJ: Max jitter (ms)</p> <p>MAXRTL: Max Round-Trip latency (ms)</p> <p>MAXFL: Max frame loss (%)</p> <p><b>Value:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the performance criteria values for the selected service.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPparameter:PERCriteria:VALue 1, LTOR,MAXJ,1000</p>

---

## SCPI Command Reference

### *Services - Profile*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VA  
Lue**

**See Also**      SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:VLAN?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VALue?**

<b>Description</b>	<p>This query returns the performance criteria values for the selected service, direction, and parameter.</p> <p>At *RST condition the values for Jitter to 2.0ms, Latency to 15.0ms &amp; Frame Loss to 0.1%.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt; SLA Parameters Performance Criteria&gt;Max Jitter(ms)Round-Trip Latency(ms)Frame Loss (%)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VALue? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, MAXJ   MAXRTL   MAXFL[, MAXimum   MINimum]</p>

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VA  
Lue?**

<b>Parameter(s)</b>	<b>Service:</b>
	<p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Type:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXJ   MAXRTL   MAXFL</p> <p>Sets the performance criteria for the selected service.</p> <p>MAXJ: Max jitter (ms)</p> <p>MAXRTL: Max Round-Trip latency (ms)</p> <p>MAXFL: Max frame loss (%)</p> <p>The program data syntax for the fourth parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current performance criteria value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VALue?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the configured performance criteria values for the selected service.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VALue 1, LTOR,MAXJ,1000 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:PERCriteria:VALue? 1, LTOR,MAXJ Returns: 1000
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:VLAN

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe:ENABLe**

<b>Description</b>	<p>This command enables or disables the type of Burst Size for selected service.</p> <p>At *RST condition CBS is enabled and EBS is disabled.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt;SLA Parameters &gt;Burst Size</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe:ENABLe &lt;wsp&gt; &lt;Service&gt;, CBS   EBS, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS   EBS</p> <p>Sets the type of Burst.</p> <p>CBS</p> <p>EBS</p> <p><b>Enable:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables the type burst size for selected service.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe:ENABLe 1, CBS,1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZE</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe:ENABLE?**

---

<b>Description</b>	<p>This query returns the status type of Burst Size for selected service.</p> <p>At *RST condition CBS is enabled and EBS is disabled.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt;SLA Parameters&gt;Burst Size</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe:ENABLE? <wsp> <Service>, CBS   EBS
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS   EBS</p> <p>Sets the type of Burst.</p> <p>CBS</p> <p>EBS</p>
<b>Response Syntax</b>	<Status>

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe:ENABle?**

<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the information rate for the selected service. 1, information rate is enabled. 0, information rate is disabled.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe:ENABle 1, CBS,1 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe:ENABle? 1, CBS Returns: 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe?

---



---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZE**

---

**Description**

This command sets the burst size for selected service and selected direction.

At \*RST condition, the value is set to 12144.

Navigation Path: Test > Setup > Profile>SLA Parameters >Burst Size>CBSEBS

**Syntax**

:SOURCE[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZE <wsp> <Service>, LTOR | RTOL, CBS | EBS, <Size> | MAXimum | MINimum

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe**

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS   EBS</p> <p>Sets the type of Burst.</p> <p>CBS</p> <p>EBS</p> <p><b>Size:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the burst size.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe 1, LTOR,cbs,2050</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZE:ENABLE</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe?**

---

**Description**

This query returns the burst size for selected service and selected direction.

At \*RST condition, the value is set to 12144.

Navigation Path: Test > Setup > Profile>SLA Parameters >Burst Size>CBSEBS

**Syntax**

:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe? <wsp> <Service>, LTOR | RTOL, CBS | EBS[, MAXimum | MINimum]

---

# SCPI Command Reference

## Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe?**

Parameter(s)	Service:
	<p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS   EBS</p> <p>Sets the type of Burst.</p> <p>CBS</p> <p>EBS</p> <p>The program data syntax for the fourth parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the burst size.</p> <p>This parameter is optional. If no token is specified, the current burst size value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Size>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe?**

<b>Response(s)</b>	<b>Size:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the burst size for selected burst size type. MAXimum, Maximum is selected as the burst size. MINimum, Minimum is selected as the burst size.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAParameter:BSIZe? 1, LTOR,cbs
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:SLAParameter:BSIZE:ENABLE? 1, CBS

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:BMRate**

<b>Description</b>	<p>This command sets the Burst max rate for selected service and selected direction.</p> <p>At *RST condition, the value is equal to Line rate.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt;Test Parameters&gt;Burst Max Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:BMRate &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, &lt;value&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Burst max rate.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:BMRate 1, LTOR,65</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARameters:NOBSequence</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:BMRate?**

<b>Description</b>	<p>This query returns the Burst max rate for selected service and selected direction.</p> <p>At *RST condition, the value is equal to Line rate.</p> <p>Navigation Path: Test &gt; Setup &gt; Profile&gt;Test Parameters&gt;Burst Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:BMRate? <wsp> <Service>, LTOR   RTOL[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Burst max rate.</p> <p>This parameter is optional. If no token is specified, the current Burst max rate value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:BMRate?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the configured Burst max rate value.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:BMRate 1, LTOR,65 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:TPARameter:BMRate? 1, LTOR Returns: 65
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARameters:NOBSeque nce?

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle**

<b>Description</b>	<p>This command enables or disables the selected service.</p> <p>This command is not associated with any *RST condition.</p> <p>Navigation Path: Test &gt; Setup &gt; Services&gt;Service</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle &lt;wsp&gt; &lt;Service&gt;, ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Enable:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the service.</p>
<b>Example(s)</b>	<code>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle 1, ON</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs</code>

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle?**

<b>Description</b>	<p>This query returns the status of the selected Service.</p> <p>This query is not associated with any *RST condition.</p> <p>Navigation Path: Test &gt; Setup &gt; Services&gt;Service</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle? &lt;wsp&gt; &lt;Service&gt;</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p>
<b>Response Syntax</b>	<p>&lt;Enable&gt;</p>
<b>Response(s)</b>	<p><b>Enable:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the service.</p> <p>1, service is enabled.</p> <p>0, service is disabled.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle 1, ON</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:PROFile:SERVice:ENABle? 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:RESDefault**

<b>Description</b>	<p>This command restores the Emix frame size CONFig to default for the selected service and selected port.</p> <p>This command is not associated with any *RST condition.</p> <p>Navigation Path: Test Configurator&gt;Setup&gt;Services&gt;Profile&gt;Frame Size(Bytes)&gt;EMIX&gt;Restore Default Button</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:RESDefault <wsp> <Service>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p>
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:RESDefault 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMESize**

<b>Description</b>	<p>This command sets the frame size for the selected service and selected Emix frame.</p> <p>This command is not associated with *RST condition.</p> <p>Navigation Path: Test Configurator&gt;Setup&gt;Services&gt;Profile&gt;Frame Size(Bytes)&gt;EMIX&gt;Emix Frame Sizes</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMESize &lt;wsp&gt; &lt;Service&gt;, &lt;EMIX frame&gt;, &lt;EMIX Frame Size&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>EMIX frame:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the EMIX Frame.</p> <p><b>EMIX Frame Size:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the EMIX Frame size.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMESize 1, 1, 64</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TEL:ETH:STR:FRAM:SIZE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMesize?**

---

<b>Description</b>	<p>This query returns the frame size for the selected service and selected Emix frame.</p> <p>This query is not associated with *RST condition.</p> <p>Navigation Path: Test Configurator&gt;Setup&gt;Services&gt;Profile&gt;Frame Size(Bytes)&gt;EMIX&gt;Emix Frame Sizes</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMesize? <wsp> <Service>, <EMIX frame>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>EMIX frame:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the EMIX Frame.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the EMIX Frame size.</p> <p>This parameter is optional. If no token is specified, the current EMIX frame size value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<FrameSize>

---

## SCPI Command Reference

### Services - Profile

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMesize?**

<b>Response(s)</b>	<b>FrameSize:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the frame size of emix frames for selected service.
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMesize 1, 1, 64 SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMesize? 1, 1 Returns: 64
<b>See Also</b>	SOURce[1..n]:DATA:TEL:ETH:STR:FRAM:SIZE

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity**

<b>Description</b>	<p>This command sets the quantity of EMIX frame for selected service.</p> <p>At *RST condition, the value is set to 2.</p> <p>Navigation Path:Test Configurator&gt;Setup&gt;Services&gt;Profile&gt;Frame Size(Bytes)&gt;EMIX&gt;Quantity</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity <wsp> <Service>, <Quantity>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Quantity:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the quantity of frames.</p>
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity 1, 5
<b>See Also</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMESize

## SCPI Command Reference

### Services - Profile

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity?

---

<b>Description</b>	<p>This query returns the quantity of EMIX frames for selected service.</p> <p>At *RST condition, the value is set to 2.</p> <p>Navigation Path:Test Configurator&gt;Setup&gt;Services&gt;Profile&gt;Frame Size(Bytes)&gt;EMIX&gt;Quantity</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity ? <wsp> <Service>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p>
<b>Response Syntax</b>	<Quantity>
<b>Response(s)</b>	<p><b>Quantity:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the quantity of EMIX frames for selected service.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity 1, 5</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity? 1</p> <p>Returns: 5</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:EMIX:FRAMesize?

---



## EtherSAM - Global

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE**

---

<b>Description</b>	<p>This command enables and disables the type of Service CONFig Test.</p> <p>At *RST condition the Ramp is enabled and Burst is disabled.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;SubTests&gt;Service CONFig Test</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE <wsp>BURSTTest   RAMPTest, ON   OFF
<b>Parameter(s)</b>	<p><b>Test:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BURSTTest   RAMPTest</p> <p>Selects the type of Service CONFig Test.</p> <p>BURSTTest: Burst test</p> <p>RAMPTest: Ramp test</p> <p><b>Enable:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Service CONFig Test.</p>
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE RAMPTest,OFF
<b>See Also</b>	SOURce[1..n][1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SERVice:ENABle

---

## SCPI Command Reference

*EtherSAM - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE?**

<b>Description</b>	<p>This query returns the status of type of Service CONFig Test.</p> <p>At *RST condition the Ramp is enabled and Burst is disabled.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;SubTests&gt;Service CONFig Test</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE? &lt;wsp&gt;BURSTTest   RAMPTest</p>
<b>Parameter(s)</b>	<p><b>Test:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BURSTTest   RAMPTest</p> <p>Selects the type of Service CONFig Test.</p> <p>BURSTTest: Burst test</p> <p>RAMPTest: Ramp test</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Service Configuration Test.</p> <p>1 Service Configuration Ramp/Burst Test Enabled</p> <p>0 Service Configuration Ramp/Burst Test Disabled</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE RAMPTest,ON</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SCOTest:TYPE? RAMPTest</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:OVERview:SERVICE:ENABLE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTest:DURation**

---

<b>Description</b>	This commands sets the duration for service performance test. At *RST condition value is 10 min. Navigation Path: Test > Setup > EtherSAM>SubTests>Service Performance Test duration
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTest:DURation <wsp> <Duration >
<b>Parameter(s)</b>	<b>Duration :</b> The program data syntax for the first parameter is defined as a <STRING PROGRAM DATA> element. Sets the service performance test Duration in ""DD:HH:MM:SS""
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ESAM:GLOB:SPRTest:DUR "10D:00:10:00"
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency?

---

## SCPI Command Reference

### *EtherSAM - Global*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTest:DURation?**

<b>Description</b>	This query returns the duration for service. At *RST condition value is 10 min. Navigation Path: Test > Setup > EtherSAM>SubTests>Service Performance Test duration
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTest:DURation?
<b>Response Syntax</b>	<Duration>
<b>Response(s)</b>	<b>Duration:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. This query returns the Global test duration estimate
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ESAM:GLOB:SPRTest:DUR "10D:00:10:00" SOUR:DATA:TEL:ETH:ESAM:GLOB:SPRTest:DUR? Returns: "10D:00:10:00"
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTTest:ENABled**

<b>Description</b>	<p>This commands enables or disables the service performance test.</p> <p>At *RST condition value is ON/OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;SubTests&gt;Service Performance Test</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTTest:ENABled &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>STATUS:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the service performance test.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ESAM:GLOB:SPRT:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:ESAM:GLOB:SPRT:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	<code>SOUR[1..n]:DATA:TEL:ETH:ESAM:GLOB:SPRTTest:DUR</code>

---

## SCPI Command Reference

*EtherSAM - Global*

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTTest:ENABled?**

---

<b>Description</b>	This query returns the status of service performance status (ON/OFF). At *RST condition value is 10 min. Navigation Path: Test > Setup > EtherSAM>SubTests>Service Performance Test duration
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:SPRTTest:ENABled?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. This query returns the service performance test status
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ESAM:GLOB:SPRT:ENAB ON SOUR:DATA:TEL:ETH:ESAM:GLOB:SPRT:ENAB? Returns: 1
<b>See Also</b>	SOUR[1..n]:DATA:TEL:ETH:ESAM:GLOB:SPRTTest:DUR?

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:TDURation:ESTimate?**

---

<b>Description</b>	This query returns the Global test duration estimate. Navigation Path: Test > Setup > EtherSAM>SubTests>Global Test duration estimation
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:TDURation:ESTimate?
<b>Response Syntax</b>	<Duration>
<b>Response(s)</b>	<b>Duration:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. This query returns the Global test duration estimate
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ESAM:GLOB:TDURation:EST?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOs:VERDict?

---

## SCPI Command Reference

### *EtherSAM - Global*

---

<b>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:PDIREction:CONFIg:STATus</b>	
<b>Description</b>	<p>This commands sets the per direction configuration status.</p> <p>At *RST condition value is true.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt; Global options &gt;Per direction configuration enable</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:PDIREction:CONFIg:STATus &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the direction configuration status.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ESAM:GLOB:PDIR:CONF:STAT ON</p>
<b>See Also</b>	<p>FETCH[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter:VERDict?</p>



**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:PDIRection:CONFIg:STATus?**

<b>Description</b>	<p>This query returns the duration for service.</p> <p>At *RST condition value is true.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt; Global options &gt;Per direction configuration enable</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:PDIRection:CONFIg:STATus?
<b>Response Syntax</b>	<Duration>
<b>Response(s)</b>	<p><b>Duration:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>This query returns the Global test duration estimate</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ESAM:GLOB:PDIR:CONF:STAT ON</p> <p>SOUR:DATA:TEL:ETH:ESAM:GLOB:PDIR:CONF:STAT?</p> <p>Returns: 1</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency:VERDIct?

---

## SCPI Command Reference

### *EtherSAM - Global*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RESTore:DEFault**

---

<b>Description</b>	This command sets the EtherSAM Configuration-Result values to default. Navigation Path: Test > Setup > EtherSAM >Global >Restore EtherSAM Defaults
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RESTore:DEFault
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:RESTore:DEFault
<b>See Also</b>	FETCH[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate:VERDict?

---

---

**:SOURce[1..n]:DATA:TELeom:ETHernet:ESAM:GLOBal:LMMode**

---

<b>Description</b>	<p>This command sets the Latency measurement mode.</p> <p>At *RST condition, the value is set to Round Trip.</p> <p>Navigation Path: Test &gt;EtherSAM &gt; Setup &gt;EtherSAM&gt; Global Options &gt; Latency Measurement Mode</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELeom:ETHernet:ESAM:GLOBal:LMMode &lt;wsp&gt;ONEWAY   RTLATENCY</code>
<b>Parameter(s)</b>	<p><b>Latency Measurement Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: ONEWAY   RTLATENCY</p> <p>Sets the Latency measurement mode.</p> <p>ONEWAY: One Way</p> <p>RTLATENCY: Round Trip Latency</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:ESAM:GLOB:LMM RTLATENCY</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELeom:ETHernet:RFC:GLOBal:LMMode?</code>

---

## SCPI Command Reference

### *EtherSAM - Global*

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:LMMode?

---

<b>Description</b>	<p>This query returns the Latency measurement mode.</p> <p>At *RST condition, the value is Round Trip.</p> <p>Navigation Path: Test &gt;EtherSAM &gt; Setup &gt;EtherSAM&gt; Global Options &gt; Latency Measurement Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:LMMode?
<b>Response Syntax</b>	<Latency measurement mode>
<b>Response(s)</b>	<p><b>Latency measurement mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Latency measurement mode.</p> <p>ONEWAY, One way is selected as latency mode.</p> <p>RTLATENCY, Round Trip Latency is selected as latency mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ESAM:GLOB:LMM RTLATENCY</p> <p>SOUR:DATA:TEL:ETH:ESAM:GLOB:LMM?</p> <p>Returns: RTLATENCY</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:LMMode?

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:GLOBal:LATency:ALARm:CURRent?**

<b>Description</b>	<p>This query returns Latency alarm current value.</p> <p>Navigation Path: Test &gt;EtherSAM&gt;Setup&gt;EtherSAM&gt;Global&gt;Global Options</p> <p>Navigation Path: Test &gt;RFC 2544&gt;Setup&gt;RFC 2544&gt;Global&gt;Global Options</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:GLOBal:LATency:ALARm:CURRent? <wsp>LOPPSL   LOPPSR
<b>Parameter(s)</b>	<p><b>Latency Measurement Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOPPSL   LOPPSR</p> <p>Selects the Loss Of 1PPS Alarm type.</p> <p>LOPPSL: Local</p> <p>LOPPSR: Remote</p>
<b>Response Syntax</b>	<Alarm Current State.>
<b>Response(s)</b>	<p><b>Alarm Current State.:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Loss Of 1PPS Alarm Current State.</p>
<b>Example(s)</b>	FETCh:DATA:TEL:ETH:ESAM:GLOBal:LATency:ALARm:CURRent? LOPPSL
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:SUMMARY:SCOTest:MAXJitter:VERDict?

## EtherSAM - Burst

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSequence**

---

<b>Description</b>	<p>This command sets the number of Burst sequence.</p> <p>At *RST condition the value is set to 2.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;Burst&gt;Number Of Burst Sequence</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSequence &lt;wsp&gt; &lt;Number of Burst Sequence&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Number of Burst Sequence:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the number of Burst sequence.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSequence 6</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:VLAN</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSequence?**

<b>Description</b>	<p>This query returns the number of burst sequence.</p> <p>At *RST condition the value is set to 2.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;Burst&gt;Number Of Burst Sequence</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSequence?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the number of Burst sequence.</p> <p>This parameter is optional. If no token is specified, the current Burst sequence value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Number of burst Sequence>
<b>Response(s)</b>	<p><b>Number of burst Sequence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of Burst sequence.</p> <p>MAXimum, returns the Burst sequence as maximum.</p> <p>MINimum, returns the Burst sequence as minimum.</p>
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:NOBSequence?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:FRASize:QUANtity

## SCPI Command Reference

### *EtherSAM - Burst*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERatio**

<b>Description</b>	<p>This command sets the refill delay ratio (%).</p> <p>At *RST condition the value is set to 50.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;Burst&gt;Refil Delay Ratio</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERatio &lt;wsp&gt; &lt;Refil delay Ratio&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Refil delay Ratio:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the refill delay ratio (%).</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERatio 50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:FRASize:QUANtity?</p>



**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERatio?**

<b>Description</b>	<p>This query returns the refill delay ratio (%).</p> <p>At *RST condition the value is set to 50.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;Burst&gt;Refil Delay Ratio</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERatio ?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the refill delay ratio (%).</p> <p>This parameter is optional. If no token is specified, the current refill delay ratio value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Refil Delay Ratio>
<b>Response(s)</b>	<p><b>Refil Delay Ratio:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the refill delay ratio (%).</p> <p>MAXimum, returns the refill delay ratio as maximum.</p> <p>MINimum, returns the refill delay ratio as minimum.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERatio 50</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:RDERatio?</p> <p>Returns: 50</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPArAmeter:INFR:ENABLE

## SCPI Command Reference

### *EtherSAM - Burst*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame**

<b>Description</b>	<p>This command sets the Burst/IR frame ratio.</p> <p>At *RST condition the value is set to 90.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;Burst&gt;Burst/ IR Frame ratio</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame &lt;wsp&gt; &lt;Burst IR frame ratio&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Burst IR frame ratio:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Burst/IR frame ratio.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame 20</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:SERVices:SLAPArAmeter:INFR:ENABLE ?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame?**

<b>Description</b>	<p>This query returns the Burst IR frame ratio.</p> <p>At *RST condition the value is set to 90.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;Burst&gt;Burst/ IR Frame ratio</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame ?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Burst/IR frame ratio.</p> <p>This parameter is optional. If no token is specified, the current Burst/IR frame ratio value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Burst ir Frame ratio>
<b>Response(s)</b>	<p><b>Burst ir Frame ratio:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Burst/IR frame ratio.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame 20</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAmeters:BIRFrame?</p> <p>Returns: 20</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:SLAPArAmeter:BSIZE?

## SCPI Command Reference

### *EtherSAM - Burst*

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:CBSt:TIME?

---

<b>Description</b>	This query returns CBS Time for mentioned Stream and direction. Navigation Path: Test > Setup > EtherSAM>Burst> CBS Time
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:CBSt:TIME? <wsp> <Service>, LTOR   RTOL
<b>Parameter(s)</b>	<b>Service:</b> The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the Service number 1 or 10. <b>Direction:</b> The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are: LTOR   RTOL Selects the Direction. (For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction) LTOR: Local to Remote RTOL: Remote to Local
<b>Response Syntax</b>	<CBS Time >
<b>Response(s)</b>	<b>CBS Time :</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns CBS Time for mentioned Stream and direction
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ESAM:BURSt:CBSt:TIME? 1, LTOR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:EBS:TIME?**

<b>Description</b>	<p>This query returns EBS Time for mentioned Stream and direction.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;Burst&gt; CBS Time</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:EBS:TIME? <wsp> <Service>, LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<EBS Time>
<b>Response(s)</b>	<p><b>EBS Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns EBS Time for mentioned Stream and direction</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ESAM:BURSt:EBS:TIME? 1, LTOR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:TXRate?

## SCPI Command Reference

### *EtherSAM - Burst*

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:TBURst:TIME?

---

<b>Description</b>	<p>This query returns Total Burst Time for mentioned Stream and direction.</p> <p>Navigation Path: Test &gt; Setup &gt; EtherSAM&gt;Burst&gt; Total Burst Time</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:BURSt:TBURst:TIME? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Total BURST time &gt;</p>
<b>Response(s)</b>	<p><b>Total BURST time :</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns Total Burst Time for mentioned Stream and direction</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ESAM:BURSt:TBURst:TIME? 1, LTOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSs?</p>

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:BURSt:TOTal?**

<b>Description</b>	This query returns the Total values for CBSEBSTotal Burst Time for specified service. Navigation Path: Test > Setup > EtherSAM>Burst> Total
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:BURSt:TOTal? <wsp>LTOR   RTOL, CBS   EBS   TBURST
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR, sets the direction from local to remote.</p> <p>RTOL, sets the direction from remote to local.</p> <p><b>Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS   EBS   TBURST</p> <p>Selects the type pf burst parameter.</p> <p>CBS: CBS</p> <p>EBS: EBS</p> <p>TBURST: Total Burst</p>
<b>Response Syntax</b>	<BURST>
<b>Response(s)</b>	<p><b>BURST:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Total values for CBSEBSTotal Burst Time for specified service.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ESAM:BURSt:TOT? LTOR,CBS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter?

## EtherSAM - Ramp

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:ADD**

---

<b>Description</b>	This command adds the step value. Navigation Path:Test Configurator>Setup>EtherSAM>Ramp>Add Button
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:ADD <wsp> <Step Value>
<b>Parameter(s)</b>	<b>Step Value:</b> The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the Step value to add
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:ADD 10
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETH:ESAM:GLOB:PDIR:CONF:STAT

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DELeTe**

---

<b>Description</b>	This command deletes the step value. Navigation Path: Test Configurator>Setup>EtherSAM>Ramp>Delete Step
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DELeTe <wsp> <Step Index>
<b>Parameter(s)</b>	<b>Step Index:</b> The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the Index of the step to be deleted
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DELeTe 2
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETH:ESAM:GLOB:PDIR:CONF:STAT?

---

## SCPI Command Reference

### *EtherSAM - Ramp*

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DEFault**

---

<b>Description</b>	This command sets the default step values. Navigation Path: Test Configurator>Setup>EtherSAM>Ramp>Default Button
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DEFault
<b>Example(s)</b>	SOURce:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DEFault
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RESTore:DEFault

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME**

<b>Description</b>	<p>This command sets the Ram Step Time.</p> <p>Navigation Path:Navigation Path:Test Setup &gt;EtherSAM&gt;Set up&gt;EtherSAM &gt; RAMP &gt;Step Time</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME &lt;wsp&gt; &lt;Ramp Step Time&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Ramp Step Time:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Set Ramp Step Time</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME 7</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOSs?</p>

---

## SCPI Command Reference

### *EtherSAM - Ramp*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME?**

---

<b>Description</b>	<p>This query returns the Ram Step Time.</p> <p>Navigation Path:Navigation Path:Test Setup &gt;EtherSAM&gt;Set up&gt;EtherSAM &gt; RAMP &gt;Step Time</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME?[&lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Returns Ramp Step Time</p> <p>This parameter is optional. If no token is specified, the current Ramp Step Time value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;RAMP STEPTIME&gt;</p>
<b>Response(s)</b>	<p><b>RAMP STEPTIME:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Ramp Step Time</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME 7</p> <p>SOURce:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME?</p> <p>Returns: 7</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter?</p>

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:DURation?**

---

<b>Description</b>	This query returns the Ram Step duration. Navigation Path:Navigation Path:Test Setup >EtherSAM>Set up>EtherSAM > RAMP >Ramp Duration
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:DURation?
<b>Response Syntax</b>	<RAMP DURATION>
<b>Response(s)</b>	<b>RAMP DURATION:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Ramp Duration
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:RAMP:DURation?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency?

---

## Labels

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel**

<b>Description</b>	<p>This command sets the path signal label (C2) for the transmitter of High Order Path (HOP).                  At *RST condition, the CONFig is set to TSiGnal.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;Labels &gt; Generated</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel &lt;wsp&gt;UNEQuipped   EQUipped   FVTMode   LOCKed   AMDS3   MDEvelopment   REServed   AM140   ATMM   MDQDb   FDDim   MHDLc   SSELf   MHLaps   SSET   M10ETHERNET   GFP   RHPPp   S1W   S2W   S3W   S4W   S5W   S6W   S7W   S8W   S9W   S10W   S11W   S12W   S13W   S14W   S15W   S16W   S17W   S18W   S19W   S20W   S21W   S22W   S23W   S24W   S25W   S26W   S27W   S28W   TSiGnal   AIStcm   AMODUK   M10FC</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel**

Parameter(s)	Label:
	The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.
	The allowed elements for this parameter are: UNEQuipped   EQUipped   FVTMode   LOCKed   AMDS3   MDEvelopment   REServed   AM140   ATMM   MDQDb   FDDim   MHDLc   SSELf   MHLaps   SSET   M10ETHERNET   GFP   RHPPp   S1W   S2W   S3W   S4W   S5W   S6W   S7W   S8W   S9W   S10W   S11W   S12W   S13W   S14W   S15W   S16W   S17W   S18W   S19W   S20W   S21W   S22W   S23W   S24W   S25W   S26W   S27W   S28W   TSIGnal   AISTcm   AMODUK   M10FC
	Selects the path signal.
	UNEQuipped: unequipped
	EQUipped: equipped non-specific
	FVTMode: floating VT (Virtual Tributary) Mode
	LOCKed: locked VT (Virtual Tributary) Mode
	AMDS3: asynchronous mapping for DS3 (Digital Signal-level 3)
	MDEvelopment: mapping under development
	REServed: reserved
	AM140: asynchronous mapping for 140M (DS4NA)
	ATMM: mapping for ATM
	MDQDb: mapping for DQDB
	FDDim: asynchronous mapping for FDDI
	MHDLc: mapping of HDLC over SONET
	SSELf: SDL with self-synchronization scrambler
	MHLaps: mapping of HDLC/LAPS
	SSET: set SDL with use of a set-reset scrambler
	M10ETHERNET: 10 Gbps ethernet (IEEE 802.3)
	GFP: Generic Framing Procedure (GFP)
	RHPPp: reserved [Obsolete High-Level Data Link Control (HDLC)/Point-to-Point Protocol (PPP) framed]
	S1W: the STS-1 w/1 VTx payload defect; S2W: the STS-1 w/2 VTx payload defect;... up to S28W: the STS-1 w/28 VTx payload defect

## SCPI Command Reference

### Labels

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel**

<b>Parameter(s)</b>	TSIGnal: the test signal, ITU-T 0.181 specific mapping. AISTcm: the STS (Synchronous Transport Signal) SPE (Synchronous Payload Envelope) AIS (Alarm Indication Signal) (TCM) AMODUK: Async Mapping of ODUK M10FC: Mapping 10 Gbits/S FC
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:HOP:PATH:LAB EQUIPPED SOUR:DATA:TEL:SDHS:HOP:PATH:LAB? Returns: EQUIPPED
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:PATtern:B16?

---



---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel?**

---

<b>Description</b>	This query returns the path signal label for the transmitter of High Order Path (HOP). At *RST condition, the CONFig is set to TSIGnal. Navigation Path: Test Setup >OTN SONET/SDH BERT OR SONET/SDH BERT> Test Configurator > Setup>Traces/Labels >Labels > Generated
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel?
<b>Response Syntax</b>	<Label>

---

## SCPI Command Reference

### Labels

---

:SOURce[1..n]:DATA:TELEcom:SDHSONet:HOP:PATH:LABel?

Response(s)	Label:
	The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.
	Returns the path signal label selected:
	UNEQUIPPED, Unequipped.
	EQUIPPED, Equipped non-specific.
	FVTMODE, Floating Virtual Tributary (VT) Mode.
	LOCKED, Locked Virtual Tributary (VT) Mode.
	AMDS3, Asynchronous mapping for Digital Signal-level 3 (DS3).
	MDEVELOPMENT, Mapping under development.
	RESERVED, Reserved.
	AM140, Asynchronous mapping for 140M (DS4NA).
	ATMM, Mapping for ATM.
	MDQDB, Mapping for DQDB.
	FDDIM, Asynchronous mapping for Fiber Distributed Data Interface (FDDI).
	MHDLC, Mapping of High-Level Data Link Control (HDLC) over SONET.
	SSELF, SDL with self-synchronization scrambler.
	MHLAPS, Mapping of High-Level Data Link Control (HDLC)/Link Access Procedure for SDH (LAPS).
	SSET, Set SDL with use of a set-reset scrambler.
	M10ETHERNET, 10 Gbps ethernet (IEEE 802.3).
	GFP, Generic Framing Procedure (GFP).
	RHPPP, Reserved [Obsolete High-Level Data Link Control (HDLC)/Point-to-Point Protocol (PPP) framed].
	S1W, STS-1 w/1 VTx payload defect.; S2W, STS-1 w/2 VTx payload defect; up to S28W, STS-1 w/28 VTx payload defect.
	TSIGNAL, Test signal, ITU-T 0.181 specific mapping path signal label.
	AISTCM, Synchronous Transport Signal (STS) Synchronous Payload Envelope (SPE) Alarm Indication Signal (AIS) (TCM) path signal label.
	AMODUK, Async Mapping of ODUK
	M10FC, Mapping 10 Gbits/S FC

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel?**

---

**Example(s)** SOUR:DATA:TEL:SDHS:HOP:PATH:LAB EQUIPPED  
SOUR:DATA:TEL:SDHS:HOP:PATH:LAB?  
Returns: EQUIPPED

**See Also** SOURce[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:PATtern:B16

---

## SCPI Command Reference

### Labels

---

---

#### :SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel:EXPEcted

---

<b>Description</b>	<p>This command sets the expected path signal label (C2) for the receiver of High Order Path (HOP).</p> <p>At *RST condition, the CONFig is set to a device-dependent value.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;Labels&gt;Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Traces &gt;Labels&gt;&gt;Expected</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel:EXPEcted &lt;wsp&gt;EQUIPPED   FVTMODE   LOCKED   AMDS3   MDEVELOPMENT   AM140   ATMM   MDQDB   FDDIM   MHDLC   SSELF   MHLAPS   SSET   M10ETHERNET   GFP   RHPPP   TSIGNAL   AMODUK   M10FC</pre>

---

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel:EXPEcted**

---

**Parameter(s)****Label:**

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: EQUIPPED | FVTMODE | LOCKED | AMDS3 | MDEVELOPMENT | AM140 | ATMM | MDQDB | FDDIM | MHDLC | SSELF | MHLAPS | SSET | M10ETHERNET | GFP | RHPPP | T SIGNAL | AMODUK | M10FC

Selects the expected path signal label.

EQUIPPED: Equipped non-specific.

FVTMODE: Floating VT (Virtual Tributary) mode.

LOCKED: Locked VT (Virtual Tributary) mode.

AMDS3: asynchronous Mapping for DS3.

MDEVELOPMENT: Mapping under Development.

AM140: asynchronous Mapping for 140M (DS4NA).

ATMM: Mapping for ATM.

MDQDB: Mapping for DQDB.

FDDIM: asynchronous Mapping for Fiber Distributed Data Interface (FDDI).

MHDLC: Mapping of HDLC (High-Level Data Link Control) over SONET.

SSELF: SDL with self-synchronization scrambler.

MHLAPS: Mapping of HDLC (High-Level Data Link Control)/LAPS (Link Access Procedure for SDH).

SSET: Set SDL with use of a set-reset scrambler.

M10ETHERNET: 10 Gbps ethernet (IEEE 802.3).

GFP: GFP (Generic Framing Procedure).

RHPPP: Reserved [obsolete HDLC (High-Level Data Link Control)/PPP (Point-to-Point Protocol) framed].

T SIGNAL: Test signal, ITU-T 0.181 specific mapping.

AMODUK: Async Mapping of ODUK

M10FC: Mapping 10 Gbits/S FC

---

## SCPI Command Reference

### *Labels*

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel:EXPEcted**

**Example(s)**      SENS:DATA:TEL:SDHS:HOP:PATH:LAB:EXP EQUIPPED  
                  SENS:DATA:TEL:SDHS:HOP:PATH:LAB:EXP?  
                  Returns: EQUIPPED

**See Also**        SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PUNeq?

---

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel:EXPeCted?**

---

<b>Description</b>	<p>This query returns the expected path signal label for receiver of High Order Path (HOP). At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Setup&gt;Traces/Labels &gt;Labels&gt;Expected</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Traces &gt;Labels&gt;&gt;Expected</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel:EXPeCted?
<b>Response Syntax</b>	<Label>

---

## SCPI Command Reference

### Labels

---

**:SENSe[1..n]:DATA:TELeom:SDHSONet:HOP:PATH:LABel:EXPEcted?**

<b>Response(s)</b>	<b>Label:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the expected path signal label. EQUIPPED, Equipped non-specific. FVTMODE, Floating Virtual Tributary (VT) mode. LOCKED, Locked Virtual Tributary (VT) mode. AMDS3, asynchronous Mapping for Digital Signal-level 3 (DS3). MDEVELOPMENT, Mapping under development. AM140, asynchronous Mapping for 140M (DS4NA). ATMM, Mapping for ATM. MDQDB, Mapping for DQDB. FDDIM, asynchronous Mapping for Fiber Distributed Data Interface (FDDI). MHDLC, Mapping of High-Level Data Link Control (HDLC) over SONET. SSELF, SDL with self-synchronization scrambler. MHLAPS, Mapping of High-Level Data Link Control (HDLC)/Link Access Procedure for SDH (LAPS). SSET, Set SDL with use of a set-reset scrambler. M10ETHERNET, 10 Gbps ethernet (IEEE 802.3). GFP, Generic Framing Procedure (GFP). RHPPP, Reserved [obsolete High-Level Data Link Control (HDLC)/Point-to-Point Protocol (PPP) framed]. TSIGNAL, Test signal, ITU-T 0.181 specific mapping. AMODUK, Async Mapping of ODUK M10FC, Mapping 10 Gbits/S FC
<b>Example(s)</b>	SENS:DATA:TEL:SDHS:HOP:PATH:LAB:EXP EQUIPPED SENS:DATA:TEL:SDHS:HOP:PATH:LAB:EXP? Returns: EQUIPPED
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:SDHSONet:HOP:PUNeq



**:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PUNeq**

<b>Description</b>	<p>This command enables or disables the Signal Label Mismatch for the expected message as well as Unequipped - Path (UNEQ-P) monitoring.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; Labels</p> <p>Navigation Path: ""Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; Sonet/SDH &gt; Labels""</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PUNeq <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Signal Label Mismatch for the expected message as well as UNEQ-P (Unequipped - Path) monitoring.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:HOP:PUN ON</p> <p>SENS:DATA:TEL:SDHS:HOP:PUN?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel:EXPEcted?

## SCPI Command Reference

### Labels

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PUNeq?**

<b>Description</b>	<p>This query returns the status of Signal Label Mismatch for the expected message as well as Unequipped - Path (UNEQ-P) monitoring.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; Traces/Labels &gt; Labels</p> <p>Navigation Path: ""Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; Sonet/SDH &gt; Labels""</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PUNeq?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Payload Label Mismatch - Path (PLM-P) / Unequipped - Path (UNEQ-P).</p> <p>1 - UNEQ-P status is enabled.</p> <p>0 - UNEQ-P status is disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:HOP:PUN ON</p> <p>SENS:DATA:TEL:SDHS:HOP:PUN?</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel:EXPEcted

## Thresholds (RFC 2544)

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:THReshold**

---

<b>Description</b>	<p>This command sets the threshold values.</p> <p>At *RST condition, this value is set to 100.0.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests&gt;Thereshold</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:THReshold &lt;wsp&gt;LATENCY   THROUGHPUT   BACKTOBACK   FRAMELOSS, LTORemote   RTOLocal   TX2RX, &lt;value&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LATENCY   THROUGHPUT   BACKTOBACK   FRAMELOSS</p> <p>Selects the type.</p> <p>LATENCY: LATENCY</p> <p>THROUGHPUT: THROUGHPUT</p> <p>BACKTOBACK: BACKTOBACK</p> <p>FRAMELOSS: FRAMELOSS</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTORemote   RTOLocal   TX2RX</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p><b>value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects maximum threshold value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:REM:THR THR,LTOR,40</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MAXRate?</p>

---

## SCPI Command Reference

### Thresholds (RFC 2544)

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:REMOte:THReshold?**

<b>Description</b>	<p>This query returns the maximum throughput rate for the throughput subtest.</p> <p>At *RST condition, this value is set to 100.0.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests&gt;Thereshold</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:REMOte:THReshold? &lt;wsp&gt;LATENCY   THROUGHPUT   BACKTOBACK   FRAMELOSS, LTORemote   RTOLocal   TX2RX[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LATENCY   THROUGHPUT   BACKTOBACK   FRAMELOSS</p> <p>Selects the type.</p> <p>LATENCY: LATENCY as type, THROUGHPUT: THROUGHPUT BACKTOBACK: BACKTOBACK FRAMELOSS: FRAMELOSS</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTORemote   RTOLocal   TX2RX</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current threshold value is returned.</p> <p>MAXimum: Biggest supported value MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:REMOte:THReshold?**

---

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the maximum threshold value.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:REM:THR THR,LTOR,40 SOUR:DATA:TEL:ETH:REM:THR? THR,LTOR Returns: 40
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MAXRate

---

## Network

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:ADDRess:IP**

<b>Description</b>	<p>This command sets the Internet Protocol (IP) address for port.</p> <p>At *RST condition, this value is set to ""10.10.0.0"".</p> <p>Navigation Path: Test &gt; Network presence &gt; IP Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:ADDRess:IP <wsp> <Address>
<b>Parameter(s)</b>	<p><b>Address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the IP address in form of string.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:ADDR:IP "10.1.1.1"</p> <p>SOUR:DATA:TEL:ETH:PORT:ADDR:IP?</p> <p>Returns: "10.1.1.1"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:ADDRess:IP?**

<b>Description</b>	<p>This query returns the Internet Protocol (IP) address for port.</p> <p>At *RST condition, this value is set to ""10.10.0.0"".</p> <p>Navigation Path: Test &gt; Smartloop Back&gt; Network presence &gt; IP Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:ADDRess:IP?
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the IP address in the form of string.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:ADDR:IP "10.1.1.1"</p> <p>SOUR:DATA:TEL:ETH:PORT:ADDR:IP?</p> <p>Returns: "10.1.1.1"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP?</p>

---

## SCPI Command Reference

### Network

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DHCP:STATus

---

<b>Description</b>	<p>This command sets the DHCP status of the Smart Loopback application.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Network Presence &gt; Automatic IP (DHCP)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DHCP:STATus <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the DHCP status for the smart loopback application.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:DHCP:STAT ON</p> <p>SOUR:DATA:TEL:ETH:NETW:DHCP:STAT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:TRANsparent:MODE:ENABle

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DHCP:STATus?**

<b>Description</b>	<p>This query returns the DHCP status of the Smart Loopback application.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Network Presence &gt; Automatic IP (DHCP)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DHCP:STATus?
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<p><b>set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the mode status.</p> <p>1, automatic IP is enabled.</p> <p>0, automatic IP is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:DHCP:STAT ON</p> <p>SOUR:DATA:TEL:ETH:NETW:DHCP:STAT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:TRANSPARENT:MODE:ENABLE?

---

## SCPI Command Reference

### Network

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK

---

<b>Description</b>	<p>This command sets the subnet mask.</p> <p>At *RST condition, this value is set to 255:255:000:000.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Network Presence &gt; Subnet Mask</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK <wsp> <address>
<b>Parameter(s)</b>	<p><b>address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the subnet mask value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:SUBN:MASK "255:255:000:000"</p> <p>SOUR:DATA:TEL:ETH:NETW:SUBN:MASK?</p> <p>Returns: "255:255:000:000"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?**

---

<b>Description</b>	<p>This query returns the subnet mask value.</p> <p>At *RST condition, this value is set to 255:255:000:000.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Network Presence &gt; Subnet Mask</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?
<b>Response Syntax</b>	<address>
<b>Response(s)</b>	<p><b>address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the subnet mask value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:SUBN:MASK "255:255:000:000"</p> <p>SOUR:DATA:TEL:ETH:NETW:SUBN:MASK?</p> <p>Returns: "255:255:000:000"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID?

---

## SCPI Command Reference

### Network

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:STATus**

<b>Description</b>	<p>This command sets the Default gateway status of the Smart Loopback application.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Network Presence &gt; Default Gateway (checkbox)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:STATus &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the Default gateway status for the smart loopback application.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:DEF:GAT:STAT ON</p> <p>SOUR:DATA:TEL:ETH:NETW:DEF:GAT:STAT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:TRANSPARENT:MODE:ENABLE</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:STATus?**

---

<b>Description</b>	<p>This query returns the Default gateway status of the Smart Loopback application.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Network Presence &gt; Default Gateway (checkbox)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:STATus?
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<p><b>set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Default gateway mode status.</p> <p>1, Default gateway is enabled.</p> <p>0, Default gateway is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:DEF:GAT:STAT ON</p> <p>SOUR:DATA:TEL:ETH:NETW:DEF:GAT:STAT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:TRANSPARENT:MODE:ENABLE?

---

## SCPI Command Reference

### Network

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:ADDRess**

<b>Description</b>	<p>This command sets the default gateway address.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Network Presence &gt; Default Gateway</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:ADDRess &lt;wsp&gt; &lt;address&gt;</p>
<b>Parameter(s)</b>	<p><b>address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Default gateway address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:DEF:GAT:ADDR "10.192.5.139"</p> <p>SOUR:DATA:TEL:ETH:NETW:DEF:GAT:ADDR?</p> <p>Returns: "10.192.5.139"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:ADDRess?**

---

<b>Description</b>	<p>This query returns the default gateway address.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Network Presence &gt; Default Gateway</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DEFault:GATeway:ADDRess?
<b>Response Syntax</b>	<address>
<b>Response(s)</b>	<p><b>address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Default gateway address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:DEF:GAT:ADDR "10.192.5.139"</p> <p>SOUR:DATA:TEL:ETH:NETW:DEF:GAT:ADDR?</p> <p>Returns: "10.192.5.139"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?

---

## SCPI Command Reference

### Network

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:MAC:ADDRess

---

<b>Description</b>	<p>This command sets the MAC address.</p> <p>At *RST condition, this value is set to ""00:00:00:00:00:00"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Network Presence &gt; MAC Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:MAC:ADDRess <wsp> <address>
<b>Parameter(s)</b>	<p><b>address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the MAC address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:MAC:ADDR "00:00:00:00:00:00"</p> <p>SOUR:DATA:TEL:ETH:NETW:MAC:ADDR?</p> <p>Returns: "00:00:00:00:00:00"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK

---



---

**:SOURce[1..n]:DATA:TELecom:ETHernet:NETWork:MAC:ADDRess?**

---

<b>Description</b>	<p>This query returns the MAC address.</p> <p>At *RST condition, this value is set to ""00:00:00:00:00:00"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; Network Presence &gt; MAC Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELecom:ETHernet:NETWork:MAC:ADDRess?
<b>Response Syntax</b>	<address>
<b>Response(s)</b>	<p><b>address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the MAC address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:MAC:ADDR "00:00:00:00:00:00"</p> <p>SOUR:DATA:TEL:ETH:NETW:MAC:ADDR?</p> <p>Returns: "00:00:00:00:00:00"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:ETHernet:NETWork:SUBNet:MASK?

---

## SCPI Command Reference

### Network

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN**

<b>Description</b>	<p>This command enables or disables VLAN type.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Network Presence &gt; Vlan Tag</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Set the VLAN for the stream.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:VLAN ON</p> <p>SOUR:DATA:TEL:ETH:NETW:VLAN?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:STACKed</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN?**

<b>Description</b>	<p>This query returns the status of VLAN type frames.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Network Presence &gt; Vlan Tag</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of VLAN type frames.</p> <p>1, VLAN is enabled.</p> <p>0, VLAN is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:VLAN ON</p> <p>SOUR:DATA:TEL:ETH:NETW:VLAN?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:STACked?

---

## SCPI Command Reference

### Network

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:STACked

---

<b>Description</b>	This command sets the value of the Virtual Local Area Network (VLAN) stacked. At *RST condition, this value is set to 1. Navigation Path: Test Setup > Test Configurator > Network Presence > Vlan Tag (dropdown)
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:STACked <wsp> <Stacked>
<b>Parameter(s)</b>	<b>Stacked:</b> The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the VLAN stacked.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:NETW:VLAN:STACked 2 SOUR:DATA:TEL:ETH:NETW:VLAN:STACked? Returns: 2
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:STACKed?**

---

<b>Description</b>	<p>This query returns the value of the VLAN stacked.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Network Presence &gt; Vlan Tag (dropdown)</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:STACKed? <wsp> <Stacked>
<b>Parameter(s)</b>	<p><b>Stacked:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the Virtual Local Area Network (VLAN) stacked.</p> <p>The value for stacked is set to 1 only.</p>
<b>Response Syntax</b>	<Stacked>
<b>Response(s)</b>	<p><b>Stacked:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the VLAN stacked.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:VLAN:STACKed 2</p> <p>SOUR:DATA:TEL:ETH:NETW:VLAN:STACKed?</p> <p>Returns: 2</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN?

---

## SCPI Command Reference

### Network

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID**

<b>Description</b>	<p>This command sets the Virtual Local Area Network (VLAN) Identification (ID).</p> <p>At *RST condition, this value is set to 2.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Network Presence &gt; VLAN ID</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID &lt;wsp&gt; &lt;Stacked&gt;, &lt;Vlanid&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Stacked:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the Virtual Local Area Network (VLAN) stacked.</p> <p>The value for stacked is set to 1 only.</p> <p><b>Vlanid:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Virtual Local Area Network (VLAN) ID.</p> <p>Choices are 0 to 4095.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:VLAN:ID 1, 1</p> <p>SOUR:DATA:TEL:ETH:NETW:VLAN:ID? 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:PRIority</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID?**

<b>Description</b>	<p>This query returns the Virtual Local Area Network (VLAN) Identification (ID).</p> <p>At *RST condition, this value is set to 2.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Network Presence &gt; VLAN ID</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID? <wsp> <Stacked>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Stacked:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the Virtual Local Area Network (VLAN) stacked.</p> <p>The value for stacked is set to 1 only.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Virtual Local Area Network (VLAN) ID.</p> <p>This parameter is optional. If no token is specified, the current VLAN ID is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Vlanid>
<b>Response(s)</b>	<p><b>Vlanid:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Virtual Local Area Network (VLAN) ID.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:VLAN:ID 1, 1</p> <p>SOUR:DATA:TEL:ETH:NETW:VLAN:ID? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:PRIority?

## SCPI Command Reference

### Network

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:TYPE**

<b>Description</b>	<p>This command selects the Ethernet of the type Virtual Local Area Network (VLAN). At *RST condition, this value is set to V8100. Navigation Path: Test Setup &gt; Test Configurator &gt; Network Presence &gt; VLAN &gt; Type</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:TYPE &lt;wsp&gt; &lt;Stacked&gt;, V8100   V88A8   V9100   V9200   V9300</p>
<b>Parameter(s)</b>	<p><b>Stacked:</b> The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element. Selects the VLAN stacked.</p> <p><b>Vtype:</b> The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element. The allowed elements for this parameter are: V8100   V88A8   V9100   V9200   V9300 Selects the VLAN type. V8100: 8100 V88A8: 88A8 V9100: 9100 V9200: 9200 V9300: 9300</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:VLAN:TYPE 1, V8100</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID:ELIGiblebit</p>



**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:TYPE?**

<b>Description</b>	This query returns the Ethernet of the type Virtual Local Area Network (VLAN). At *RST condition, this value is set to V8100. Navigation Path: Test Setup > Test Configurator > Network Presence > VLAN > Type
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:TYPE? <wsp> <Stacked>
<b>Parameter(s)</b>	<b>Stacked:</b> The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the VLAN stacked.
<b>Response Syntax</b>	<Vtype>
<b>Response(s)</b>	<b>Vtype:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of the VLAN. V8100, the VLAN type 8100 is selected. V88A8, the VLAN type 88A8 is selected. V9100, the VLAN type 9100 is selected. V9200, the VLAN type 9200 is selected. V9300, the VLAN type 9300 is selected.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:NETW:VLAN:TYPE 1, V8100 SOUR:DATA:TEL:ETH:NETW:VLAN:TYPE? 1 Returns: V8100
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID:ELIGiblebit?

## SCPI Command Reference

### Network

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:PRIority

---

<b>Description</b>	<p>This command sets the Virtual Local Area Network (VLAN) user priority.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Network Presence &gt; VLAN &gt; Priority</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:PRIority &lt;wsp&gt; &lt;Stacked&gt;, &lt;VlanPriority&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Stacked:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the Virtual Local Area Network (VLAN) stacked.</p> <p>The value for stacked is set to 1 only.</p> <p><b>VlanPriority:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Virtual Local Area Network (VLAN) priority. Choices are 0 to 7.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:NETW:VLAN:PRI 1, 1 SOUR:DATA:TEL:ETH:NETW:VLAN:PRI? 1 Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID</pre>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:PRIority?**

<b>Description</b>	<p>This query returns the Virtual Local Area Network (VLAN) user priority.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Network Presence &gt; VLAN &gt; Priority</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:PRIority? <wsp> <Stacked>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Stacked:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Select the Virtual Local Area Network (VLAN) stacked.</p> <p>The value for stacked is set to 1 only.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current VLAN priority is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Priority>
<b>Response(s)</b>	<p><b>Priority:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Virtual Local Area Network (VLAN) user priority.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:VLAN:PRI 1, 1</p> <p>SOUR:DATA:TEL:ETH:NETW:VLAN:PRI? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID?

## SCPI Command Reference

### Network

---

<b>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID:ELIGiblebit</b>	
<b>Description</b>	<p>This command enables or disables the Virtual Local Area Network (VLAN) Identification (ID) eligible bit.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; Network Presence &gt; VLAN &gt; Eligible bit</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID:ELIGiblebit &lt;wsp&gt; &lt;Stacked&gt;, ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Stacked:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Enables or disables the Virtual Local Area Network (VLAN) ID stacked.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the eligible bit for the specific VLAN (Virtual Local Area Network) ID (Identification).</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:NETW:VLAN:ID:ELIG 1, ON SOUR:DATA:TEL:ETH:NETW:VLAN:ID:ELIG? 1</pre> <p>Returns: 1</p>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:TYPE</code>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID:ELIGiblebit?**

<b>Description</b>	<p>This query returns the status of Virtual Local Area Network (VLAN) Identification (ID) eligible bit.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: TestSetup &gt; Test CONFig - - &gt; network presence - &gt; VLAN - &gt; Eligible bit</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:ID:ELIGiblebit? <wsp> <Stacked>
<b>Parameter(s)</b>	<p><b>Stacked:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Enables or disables the Virtual Local Area Network (VLAN) ID stacked.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Virtual Local Area Network (VLAN) ID eligible bit.</p> <p>1, eligible bit is enabled.</p> <p>0, eligible bit is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:VLAN:ID:ELIG 1, ON</p> <p>SOUR:DATA:TEL:ETH:NETW:VLAN:ID:ELIG? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:VLAN:TYPE?

## SCPI Command Reference

### Network

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault

---

<b>Description</b>	<p>This command Allows to activate/deactivate the display of a pop-up upon start-up of the module asking to restore the factory settings or not. The last configuration settings are used when disabled.</p> <p>At *RST condition, this value is set to AUTOamtic.</p> <p>Navigation Path: Test &gt; Test Configurator&gt;Network Presence &gt; MAC &gt; Factory default</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Allows to activate/deactivate the display of a pop-up upon start-up of the module asking to restore the factory settings or not</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault OFF</p> <p>SOURce:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:GLOB:IPV:MODE

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault?**

---

<b>Description</b>	<p>This query returns the Factory default state.</p> <p>At *RST condition, this value is set to AUTOamtic.</p> <p>Navigation Path: Test &gt; Test Configurator&gt;Network Presence &gt; MAC &gt; Factory default</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Factory Default State</p> <p>1, Returns factory default is enable</p> <p>0, Returns factory default is disable</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault ON</p> <p>SOURce:DATA:TELEcom:ETHernet:NETWork:FACTory:DEFault?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:GLOB:IPV:MODE?

---

## SCPI Command Reference

### Network

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DATalink:TYPE?**

<b>Description</b>	<p>This returns the frame format selected.</p> <p>At *RST condition, this value is set to ETHernet.</p> <p>Navigation Path: BERT &gt; TestSetup &gt; Test Configurator &gt; Network Presence &gt; MAC &gt; Frame format</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DATalink:TYPE?
<b>Response Syntax</b>	<Datalink>
<b>Response(s)</b>	<p><b>Datalink:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the status of Frame Format</p> <p>ETHERNETII, Returns ETHERNETII as datalink.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:NETWork:DATalink:TYPE ETHERNETII</p> <p>SOURce:DATA:TELEcom:ETHernet:NETWork:DATalink:TYPE?</p> <p>Returns: ETHERNETII</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:DATalink?

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCal:IPVersion:ADDRESS**

---

<b>Description</b>	<p>This command sets the Link-Local IPv6 Address.</p> <p>At *RST condition, this value is set to ""FE80:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig &gt; Source Link-Local IPv6 Address.</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAl:IPVersion:ADDRESS &lt;wsp&gt; &lt;address&gt;</code>
<b>Parameter(s)</b>	<p><b>address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Sets the Local IPv6 Address</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:NETW:IPV IPV6 SOUR:DATA:TEL:ETH:NETW:LOC:IPV:ADDR FE80:0000:0000:0000:0000:0000:0000:1111 SOUR:DATA:TEL:ETH:NETW:LOC:IPV:ADDR? Returns: FE80:0000:0000:0000:0000:0000:0000:1111</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?</code>

---

## SCPI Command Reference

### Network

---

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAl:IPVersion:ADDRess?**

---

<b>Description</b>	<p>This query returns the Link-Local IPv6 Address.</p> <p>At *RST condition, this value is set to ""FE80:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig &gt; Source Link-Local IPv6 Address.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAl:IPVersion:ADDRess?
<b>Response Syntax</b>	<address>
<b>Response(s)</b>	<p><b>address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Link-Local IPv6 Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:IPV IPV6</p> <p>SOUR:DATA:TEL:ETH:NETW:LOC:IPV:ADDR</p> <p>FE80:0000:0000:0000:0000:0000:0000:1111</p> <p>SOUR:DATA:TEL:ETH:NETW:LOC:IPV:ADDR?</p> <p>Returns: FE80:0000:0000:0000:0000:0000:0000:1111</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?

---

## RFC 2544 - Subtests

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MAXRate**

---

<b>Description</b>	<p>This command sets the maximum throughput rate for the throughput subtest.</p> <p>At *RST condition, this value is set to 100.0.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MAXRate &lt;wsp&gt;TX2RX   LTORemote   RTOLocal, &lt;Maxrate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <p><b>Maxrate:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects maximum throughput rate.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:MAXR TX2RX,20.00</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:MAXR? TX2RX</p> <p>Returns: 20.00</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZE</p>

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:MAXRate?

---

<b>Description</b>	<p>This query returns the maximum throughput rate for the throughput subtest.</p> <p>At *RST condition, this value is set to 100.0.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:MAXRate? <wsp>TX2RX   LTORemote   RTOLocal   [, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal  </p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current throughput rate for the throughput subtest is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Maxrate>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MAXRate?**

<b>Response(s)</b>	<b>Maxrate:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the maximum throughput rate for the test.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:THR:MAXR TX2RX,10.00 SOUR:DATA:TEL:ETH:RFC:THR:MAXR? TX2RX Returns: 10.00
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FSIZE?

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ACCuracy

---

<b>Description</b>	<p>This command sets the accuracy value for the throughput subtest.</p> <p>*RST values are 1.0 for Gbit/s, 1000 for Mbit/s and 1.0 for %.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ACCuracy &lt;wsp&gt; &lt;Accuracy&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Accuracy:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the throughput accuracy value.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:RFC:THR:ACC 10.00 SOUR:DATA:TEL:ETH:RFC:THR:ACC? Returns: 10.00</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MAXRate</pre>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ACCuracy?**

<b>Description</b>	<p>This query returns the accuracy value for the throughput subtest.</p> <p>*RST values are 1.0 for Gbit/s, 1000 for Mbit/s and 1.0 for %.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:ACCuracy?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current accuracy value for the throughput subtest is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Accuracy>
<b>Response(s)</b>	<p><b>Accuracy:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the accuracy measurement value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:ACC 10.00</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:ACC?</p> <p>Returns: 10.00</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MAXRate?

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:AERRors

---

<b>Description</b>	<p>This command sets the number of acceptable errors for the throughput subtest.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:AERRors &lt;wsp&gt; &lt;Errors&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Errors:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the number of acceptable throughput errors. Choices are 0 to 10.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:AERR 10</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:AERR?</p> <p>Returns: 10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors</p>

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:AERRors?**

<b>Description</b>	<p>This query returns the number of acceptable errors for the throughput subtest.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:AERRors?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current number of acceptable errors for the throughput subtest is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Errors>
<b>Response(s)</b>	<p><b>Errors:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of acceptable errors for the throughput subtest.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:AERR 10</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:AERR?</p> <p>Returns: 10</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors?

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TAVerage

---

<b>Description</b>	<p>This command sets the number of times the throughput subtest is generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TAVerage &lt;wsp&gt; &lt;Average&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Average:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the number of throughput trials. Choices are 1 to 50.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:TAV 10</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:TAV?</p> <p>Returns: 10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TAVerage?**

<b>Description</b>	<p>This query returns the number of times the throughput subtest is generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TAVerage?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current number of times the throughput subtest generated is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Average>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of times the throughput test is generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:TAV 10</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:TAV?</p> <p>Returns: 10</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:VALidations

---

<b>Description</b>	<p>This command sets the number of times the throughput result should be validated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:VALidations &lt;wsp&gt; &lt;Validations&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Validations:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the number of throughput validations. Choices are 1 to 50.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:VAL 10</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:VAL?</p> <p>Returns: 10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:VALidations?**

<b>Description</b>	<p>This query returns the number of times the result should be validated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:VALidations?[ &lt;wsp&gt;MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current number of times the result should be validated is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Validations&gt;</code>
<b>Response(s)</b>	<p><b>Validations:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of times the result should be validated.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:RFC:THR:VAL 10 SOUR:DATA:TEL:ETH:RFC:THR:VAL? Returns: 10</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors?</code>

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TTIME

---

<b>Description</b>	<p>This command sets the trial duration for the throughput subtest.</p> <p>At *RST condition, this value is set to ""00:01"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TTIME <wsp> <Time>
<b>Parameter(s)</b>	<p><b>Time:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the throughput test time value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:TTIM "30:00"</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:TTIM?</p> <p>Returns: "30:00"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TTIME?**

<b>Description</b>	<p>This query returns the trial duration for the throughput subtest.</p> <p>At *RST condition, this value is set to ""00:01"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TTIME?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the trial duration for the test.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:THR:TTIM "30:00"</p> <p>SOUR:DATA:TEL:ETH:RFC:THR:TTIM?</p> <p>Returns: "30:00"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors?

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames

---

<b>Description</b>	<p>This command sets the burst of frames (Max. time worth of frames/Burst Time) sent with minimum inter-frame gaps to the device under test and the number of forwarded frames counted.</p> <p>At *RST condition, this value is set to ""00:01"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames <wsp> <Time>
<b>Parameter(s)</b>	<p><b>Time:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the burst time.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:BCKT:MTFR "00:01"</p> <p>SOUR:DATA:TEL:ETH:RFC:BCKT:MTFR?</p> <p>Returns: "00:01"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy</p>

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames?**

<b>Description</b>	<p>This query returns the burst of frames (Max. time worth of frames/Burst Time) sent with minimum inter-frame gaps to the device under test and the number of forwarded frames counted.</p> <p>At *RST condition, this value is set to ""00:01"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the maximum time worth of frames (Burst Time).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:BCKT:MTFR "00:01"</p> <p>SOUR:DATA:TEL:ETH:RFC:BCKT:MTFR?</p> <p>Returns: "00:01"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy</p>

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy

---

<b>Description</b>	<p>This command sets the accuracy measurement value in frames.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy &lt;wsp&gt; &lt;Accuracy&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Accuracy:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects accuracy measurement value in frames. Choices are 1 to 50.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:RFC:BCKT:ACC 10 SOUR:DATA:TEL:ETH:RFC:BCKT:ACC? Returns: 10</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames?</pre>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy?**

<b>Description</b>	<p>This query returns the accuracy measurement value in frames.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:ACCuracy?[\n&lt;wsp&gt;MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current accuracy measurement in frames is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Accuracy&gt;</code>
<b>Response(s)</b>	<p><b>Accuracy:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the accuracy measurement value.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:RFC:BCKT:ACC 10 SOUR:DATA:TEL:ETH:RFC:BCKT:ACC? Returns: 10</pre>
<b>See Also</b>	<p><code>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames</code></p> <p><code>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:MTFRames?</code></p>

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors

---

<b>Description</b>	<p>This command sets the number of acceptable errors for the back-to-back subtest.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors &lt;wsp&gt; &lt;Errors&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Errors:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the number of acceptable errors. Choices are 0 to 10.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:BCKT:AERR 10</p> <p>SOUR:DATA:TEL:ETH:RFC:BCKT:AERR?</p> <p>Returns: 10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:AERRors</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:AERRors?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors?**

<b>Description</b>	<p>This query returns the number of acceptable errors for the back-to-back subtest.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:AERRors?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current number of acceptable errors for the back-to-back subtest is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Errors>
<b>Response(s)</b>	<p><b>Errors:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of acceptable errors.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:BCKT:AERR 10</p> <p>SOUR:DATA:TEL:ETH:RFC:BCKT:AERR?</p> <p>Returns: 10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:AERRors</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:AERRors?</p>

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage

---

<b>Description</b>	<p>This command sets the number of times the back-to-back subtest is generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage &lt;wsp&gt; &lt;Average&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Average:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the number of trials. Choices are 1 to 100.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:RFC:BCKT:TAV 10 SOUR:DATA:TEL:ETH:RFC:BCKT:TAV? Returns: 10</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TAVerage SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TAVerage?</pre>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage?**

<b>Description</b>	<p>This query returns the number of times the back-to-back subtest is generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage?[ &lt;wsp&gt;MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current number of times the back-to-back subtest is generated is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Average&gt;</code>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of trials.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:RFC:BCKT:TAV 10 SOUR:DATA:TEL:ETH:RFC:BCKT:TAV? Returns: 10</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TAVerage SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TAVerage?</pre>

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:NBURst?**

<b>Description</b>	<p>This query returns the number of bursts that is generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:NBURst?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current number of bursts generated is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Burst&gt;</p>
<b>Response(s)</b>	<p><b>Burst:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of bursts.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:BCKT:NBUR?</p> <p>Returns: 10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:NBURst?</p>



**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:TTIME**

<b>Description</b>	<p>This command sets the trial duration for the Frame Loss subtest.</p> <p>At *RST condition, this value is set to ""00:01"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:TTIME <wsp> <Time>
<b>Parameter(s)</b>	<p><b>Time:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the trial duration for the test.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FLOS:TTIM "30:00"</p> <p>SOUR:DATA:TEL:ETH:RFC:FLOS:TTIM?</p> <p>Returns: "30:00"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TTIME</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TTIME?</p>

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:TTIME?

---

<b>Description</b>	<p>This query returns the trial duration for the Frame Loss subtest.</p> <p>At *RST condition, this value is set to ""00:01"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:TTIME?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the test trial duration value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FLOS:TTIM "30:00"</p> <p>SOUR:DATA:TEL:ETH:RFC:FLOS:TTIM?</p> <p>Returns: "30:00"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TTIME</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TTIME?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:TGRanularity**

<b>Description</b>	<p>This command sets the Frame Loss subtest granularity percentage.</p> <p>At *RST condition, this value is set to 10.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:TGRanularity &lt;wsp&gt; &lt;Granularity&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Granularity:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the test granularity percentage. Choices are 1 to 10.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:RFC:FLOS:TGR 10 SOUR:DATA:TEL:ETH:RFC:FLOS:TGR? Returns: 10</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:TTIME?</pre>

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:TGRanularity?

---

<b>Description</b>	<p>This query returns the Frame Loss subtest granularity percentage.</p> <p>At *RST condition, this value is set to 10.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:TGRanularity?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current percentage of Frame Loss subtest granularity is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Granularity>
<b>Response(s)</b>	<p><b>Granularity:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the test granularity percentage.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FLOS:TGR 10</p> <p>SOUR:DATA:TEL:ETH:RFC:FLOS:TGR?</p> <p>Returns: 10</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:TTime

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:TAVerage**

<b>Description</b>	<p>This command sets the number of times the Frame Loss subtest is generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:TAVerage <wsp> <Average>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Average:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the number of times the Frame Loss test is generated. Choices are 1 to 50.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FLOS:TAV 10</p> <p>SOUR:DATA:TEL:ETH:RFC:FLOS:TAV?</p> <p>Returns: 10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage?</p>

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:TAVerage?

---

<b>Description</b>	<p>This query returns the number of times the Frame Loss subtest is generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:TAVerage? [ <wsp>MAXimum   MINimum ]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the number of times the Frame Loss test is generated. Choices are 1 to 50.</p> <p>This parameter is optional. If no token is specified, the current number of times the Frame Loss subtest generated is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Average>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of times the Frame Loss test is generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FLOs:TAV 10</p> <p>SOUR:DATA:TEL:ETH:RFC:FLOs:TAV?</p> <p>Returns: 10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:MAXRate**

<b>Description</b>	<p>This command sets the maximum rate for the Frame Loss subtest.</p> <p>At *RST condition, this value is set to 100.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:MAXRate <wsp>TX2RX   LTORemote   RTOLocal, <Maxrate>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <p><b>Maxrate:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the maximum rate for the Frame Loss test.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:FLOS:MAXR TX2RX,10.00</p> <p>SOUR:DATA:TEL:ETH:RFC:FLOS:MAXR? TX2RX</p> <p>Returns: 10.00</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:MAXRate</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THROUGHput:MAXRate?</p>

## SCPI Command Reference

### RFC 2544 - Subtests

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:MAXRate?**

<b>Description</b>	<p>This query returns the maximum rate for the Frame Loss subtest.</p> <p>At *RST condition, this value is set to 100.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSS:MAXRate? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the maximum rate for the Frame Loss test.</p> <p>This parameter is optional. If no token is specified, the current maximum rate for the Frame Loss subtest is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Maxrate&gt;</p>

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:MAXRate?**

<b>Response(s)</b>	<b>Maxrate:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the maximum rate for the Frame Loss test.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:FLOS:MAXR TX2RX,10.00 SOUR:DATA:TEL:ETH:RFC:FLOS:MAXR? TX2RX Returns: 10.00
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MAXRate SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:MAXRate?

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TAVerage**

<b>Description</b>	<p>This command sets the number of times the latency subtest is generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TAVerage &lt;wsp&gt; &lt;Average&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Average:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the number of times the latency test is generated.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:TAV 10</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:TAV?</p> <p>Returns: 10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:TTIME</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TAVerage?**

<b>Description</b>	<p>This query returns the number of times the latency subtest is generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TAVerage?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the number of times the latency test is generated.</p> <p>This parameter is optional. If no token is specified, the current number of times the latency subtest generated is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Average>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of times the latency test is generated.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:TAV 10</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:TAV?</p> <p>Returns: 10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TAVerage?</p>

## SCPI Command Reference

### RFC 2544 - Subtests

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate[1..n]**

<b>Description</b>	<p>This command selects the maximum rate for the latency subtest.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC &gt; Configure Per Frame Size</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate[1..n] &lt;wsp&gt;TX2RX   LTORemote   RTOLocal, &lt;Maxrate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <p><b>Maxrate:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the maximum rate for the Latency test.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:MAXR1 TX2RX,10.00</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:MAXR1? TX2RX</p> <p>Returns: 10.00</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate1?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate[1..n]?**

<b>Description</b>	<p>This query returns the maximum rate for the latency subtest.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC &gt; Configure Per Frame Size</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate[1..n]? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the maximum rate for the Latency test.</p> <p>This parameter is optional. If no token is specified, the current maximum rate for the latency subtest is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Maxrate&gt;</p>

---

## SCPI Command Reference

*RFC 2544 - Subtests*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate[1..n]?**

<b>Response(s)</b>	<b>Maxrate:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the maximum rate for the latency test.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:LAT:MAXR1 TX2RX,10.00 SOUR:DATA:TEL:ETH:RFC:LAT:MAXR1? TX2RX Returns: 10.00
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate5

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPItest**

<b>Description</b>	<p>This command enables or disables the copy from the throughput test to get values from the throughput test results.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPItest <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the copy from the throughput test.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:COPI ON</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:COPI?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPItest?

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPYtest?

---

<b>Description</b>	<p>This query returns the status of copy from the throughput test to get values from the throughput test results.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPYtest?
<b>Response Syntax</b>	<Get>
<b>Response(s)</b>	<p><b>Get:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of copy from the throughput test.</p> <p>0, returns the status of copy from the throughput test as OFF.</p> <p>1, returns the status of copy from the throughput test as ON.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:COPY ON</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:COPY?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPYtest

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate:SETall**

---

<b>Description</b>	<p>This command sets the value for all frame sizes.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC &gt; Configure Per Frame Size</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate:SETall &lt;wsp&gt;TX2RX   LTORemote   RTOLocal, &lt;Maxrate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <p><b>Maxrate:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value for all frame sizes. Choices varies according to the selection of unit.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:MAXR:SET TX2RX,50.00</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:MAXR:SET? TX2RX</p> <p>Returns: 50.00</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPY</p>

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

**:SOURce[1..n]:DATA:TELecom:ETHernet:RFC:LAency:MAXRate:SETall?**

<b>Description</b>	<p>This query returns the value for all frame sizes.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC &gt; Configure Per Frame Size</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELecom:ETHernet:RFC:LAency:MAXRate:SETall? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value for all frame sizes. Choices varies according to the selection of unit.</p> <p>This parameter is optional. If no token is specified, the current value for all frame sizes is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Maxrate&gt;</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MAXRate:SETall?**

<b>Response(s)</b>	<b>Maxrate:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the value for all frame sizes.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:LAT:MAXR:SET TX2RX,50.00 SOUR:DATA:TEL:ETH:RFC:LAT:MAXR:SET? TX2RX Returns: 50.00
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:COPY?

---

## SCPI Command Reference

### RFC 2544 - Subtests

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MARGin**

<b>Description</b>	<p>This command sets the maximum rate for each frame size.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests &gt; Latency</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MARGin &lt;wsp&gt; &lt;Margin&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Margin:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the maximum rate for each frame size.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:MARG 1</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:MARG?</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:TTIME</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MARGin?**

<b>Description</b>	<p>This query returns the maximum rate for each frame size.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests &gt; Latency</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:MARGin?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the maximum rate for each frame size.</p> <p>This parameter is optional. If no token is specified, the current maximum rate for each frame size is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Percentage>
<b>Response(s)</b>	<p><b>Percentage:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the maximum rate for each frame size.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:LAT:MARG?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:TTime?

## SCPI Command Reference

### RFC 2544 - Subtests

---

**:SOURce[1..n]:DATA:TELecom:ETHernet:RFC:LATency:TTIME**

<b>Description</b>	<p>This command sets the trial duration for the latency subtest.</p> <p>At *RST condition, this value is set to 00:01.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELecom:ETHernet:RFC:LATency:TTIME &lt;wsp&gt; &lt;Time&gt;</code>
<b>Parameter(s)</b>	<p><b>Time:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the trial duration for the latency test.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:TTIM "01:00"</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:TTIM?</p> <p>Returns: "01:00"</p>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELecom:ETHernet:RFC:FLOs:s:MARGin</code>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TTIME?**

---

<b>Description</b>	<p>This query returns the trial duration value for the latency subtest.</p> <p>At *RST condition, this value is set to ""00:01"".</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt; Subtests</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TTIME?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the trial duration for the latency test.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:LAT:TTIM "01:00"</p> <p>SOUR:DATA:TEL:ETH:RFC:LAT:TTIM?</p> <p>Returns: "01:00"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:MARGin?

---

## SCPI Command Reference

RFC 2544 - Subtests

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:LMMode**

<b>Description</b>	<p>This command sets the Latency measurement mode.</p> <p>At *RST condition, the value is set to Round Trip.</p> <p>Navigation Path: Test &gt; Ether SAM &gt; Setup &gt; RFC &gt; Global Options &gt; Latency Measurement Mode</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:LMMode &lt;wsp&gt;ONEWAY   RTLATENCY</p>
<b>Parameter(s)</b>	<p><b>Latency Measurement Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: ONEWAY   RTLATENCY</p> <p>Sets the Latency measurement mode.</p> <p>ONEWAY: One Way</p> <p>RTLATENCY: Round Trip Latency</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:GLOB:LMM RTLATENCY</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:LMMode?</p>



**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:LMMode?**

<b>Description</b>	<p>This query returns the Latency measurement mode.</p> <p>At *RST condition, the value is Round Trip.</p> <p>Navigation Path: Test &gt; Ether SAM &gt; Setup &gt; RFC &gt; Global Options &gt; Latency Measurement Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:LMMode?
<b>Response Syntax</b>	<Latency measurement mode>
<b>Response(s)</b>	<p><b>Latency measurement mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Latency measurement mode.</p> <p>ONEWAY, One way is selected as latency mode.</p> <p>RTLATENCY, Round Trip Latency is selected as latency mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:GLOB:LMM RTLATENCY</p> <p>SOUR:DATA:TEL:ETH:RFC:GLOB:LMM?</p> <p>Returns: RTLATENCY</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:GLOBal:LMMode

---

## IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAl:IPVersion:MODE**

---

<b>Description</b>	<p>This command selects the Link-Local IPv6 Mode.</p> <p>At *RST condition, this value is set to STATic.</p> <p>Navigation Path: Test &gt;Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig&gt;Link-Local IPv6 Address&gt;Mode</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAl:IPVersion:MODE &lt;wsp&gt;SAUTO   STATic</p>
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SAUTO   STATic</p> <p>Sets the Local IPv6 Mode.</p> <p>SAUTO, sets mode to stateless Auto</p> <p>STATic, sets mode to static</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:LOC:IPV:MODE STAT</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAl:IPVersion:MODE?**

<b>Description</b>	<p>This query returns the Link-Local IPv6 Mode.</p> <p>At *RST condition, this value is set to STATic.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFIg &gt;IP&gt;Link-Local IPv6 Address&gt;Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:LOCAl:IPVersion:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Link-Local IPv6 mode.</p> <p>STATic, Static is selected as Link-Local IPv6 mode.</p> <p>SAUTo, Stateless Auto. (SAUTo) is selected as Link-Local IPv6 mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:LOC:IPV:MODE STAT</p> <p>SOUR:DATA:TEL:ETH:NETW:LOC:IPV:MODE?</p> <p>Returns: STATic</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?

---

## SCPI Command Reference

### IPv6 Address Configuration

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:ADDRESS

---

<b>Description</b>	<p>This command sets the Global IPv6 Address.</p> <p>At *RST condition, this value is set to ""2001:0000:0000:0000.0000:0000:0000.1111"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig &gt;Global IPv6 Address.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:ADDRESS <wsp> <address>
<b>Parameter(s)</b>	<p><b>address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Sets the Global IPv6 Address</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:MODE STAT SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:IIC OFF SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:ADDR "2001:0000:0000:0000:0000:0000:0000:0000" SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:ADDR? Returns: "2001:0000:0000:0000:0000:0000:0000:0000"</pre>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:ADDRess?**

---

<b>Description</b>	<p>This query returns the Global IPv6 Address.</p> <p>At *RST condition, this value is set to ""2001:0000:0000:0000:0000:0000:1111"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFIg&gt;Global IPv6 Address.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:ADDRess?
<b>Response Syntax</b>	<address>
<b>Response(s)</b>	<p><b>address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Global IPv6 Address.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:MODE STAT SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:IIC OFF SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:ADDR "2001:0000:0000:0000:0000:0000:0000" SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:ADDR? Returns: "2001:0000:0000:0000:0000:0000:0000:0000"</pre>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?

---

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:MODE**

<b>Description</b>	<p>This command selects the Global IPv6 Mode.</p> <p>At *RST condition, this value is set to NONE.</p> <p>Navigation Path: Test &gt;Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig &gt;Global IPv6 Address &gt; Mode.</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:MODE &lt;wsp&gt;STATIC   SAUTO   NONE</p>
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: STATIC   SAUTO   NONE</p> <p>Sets the Global IPv6 Mode.</p> <p>SAUTO,sets mode to stateless Auto</p> <p>STATic, sets mode to static</p> <p>NONE, sets mode to None</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:MODE STAT</p> <p>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:MODE?</p> <p>Returns: STATIC</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:MODE?**

---

<b>Description</b>	<p>This query returns the Global IPv6 Mode.</p> <p>At *RST condition, this value is set to NONE.</p> <p>Navigation Path: Test &gt;Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig&gt;Global IPv6 Address &gt; Mode.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Global IPv6 mode.</p> <p>NONE, No Global IPv6 mode is selected.</p> <p>STATIC, Static is selected as Global IPv6 mode.</p> <p>SAUTO, Stateless Auto. (SAUTO) is selected as Global IPv6 mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:MODE STAT</p> <p>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:MODE?</p> <p>Returns: STATIC</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:SUBNet:MASK?

---

## SCPI Command Reference

### *IPv6 Address Configuration*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:PMASK**

<b>Description</b>	<p>This command sets the Global IPv6 Address Prefix Mask.</p> <p>At *RST condition, this value is set to ""0000:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig &gt;Global IPv6 Address &gt; Prefix Mask</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:PMASK &lt;wsp&gt; &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Global IPv6 Address Prefix Mask.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:MODE STAT</p> <p>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:PMAS</p> <p>"FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000"</p> <p>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:PMAS?</p> <p>Returns: "FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:LOCal:IPV:MODE?</p>



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:PMASK?**

<b>Description</b>	<p>This query returns the Global IPv6 Address Prefix Mask.</p> <p>At *RST condition, this value is set to ""0000:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig&gt;Global IPv6 Address &gt; Prefix Mask</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:PMASK?
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Global IPv6 Address Prefix Mask.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:MODE STAT SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:PMAS "FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000" SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:PMAS? Returns: "FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000"</pre>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:LOCAl:IPV:MODE

---

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:IICoupled**

<b>Description</b>	<p>This command enables or disables the Interface ID of the Global IPv6 address.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt;Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig &gt;Global IPv6 Address &gt;Interface ID Coupled.</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:IICoupled &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Interface ID of the Global IPv6 address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:MODE STAT</p> <p>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:IIC OFF</p> <p>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:IIC?</p> <p>Returns: 0</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:GLOB:IPV:ADDRess?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:GLOB:IPV:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:GLOB:IPV:IICoupled</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:IICoupled?**

---

<b>Description</b>	<p>This query returns the status of Interface ID of the Global IPv6 address.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFIg&gt;Global IPv6 Address &gt;Interface ID Coupled.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:GLOBal:IPVersion:IICoupled?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Interface ID of the Global IPv6 address.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:MODE STAT SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:IIC OFF SOUR:DATA:TEL:ETH:NETW:GLOB:IPV:IIC? Returns: 0</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:GLOB:IPV:ADDRess SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:GLOB:IPV:MODE SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:GLOB:IPV:IICoupled</pre>

---

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:ADDRess**

<b>Description</b>	<p>This command sets the Default Gateway Address of IPv6.</p> <p>At *RST condition, this value is set to ""FE80:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig&gt;Default Gateway&gt;Address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:ADDRess &lt;wsp&gt; &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Default Gateway IPv6 Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:DGAT:IPV:MODE STATIC SOUR:DATA:TEL:ETH:NETW:DGAT:IPV:ADDR "FE80:0000:0000:0000:0000:0000:0000:0000" SOUR:DATA:TEL:ETH:NETW:DGAT:IPV:ADDR? Returns: "FE80:0000:0000:0000:0000:0000:0000:0000"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:DGATeway:IPV:ADDRess SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:DGATeway:IPV:MODE</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:ADDRess?**

---

<b>Description</b>	<p>This query returns the Default Gateway Address of IPv6.</p> <p>At *RST condition, this value is set to ""FE80:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt;Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig &gt;Default Gateway&gt;Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:ADDRess?
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Default Gateway IPv6 Address.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:NETW:DGAT:IPV:MODE STATIC SOUR:DATA:TEL:ETH:NETW:DGAT:IPV:ADDR "FE80:0000:0000:0000:0000:0000:0000:0000" SOUR:DATA:TEL:ETH:NETW:DGAT:IPV:ADDR? Returns: "FE80:0000:0000:0000:0000:0000:0000:0000"</pre>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:DGATeway:IPV:MODE?

---

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:MODE**

<b>Description</b>	<p>This command selects the Default Gateway IPv6 Address Mode.</p> <p>At *RST condition, this value is set to AUTOMATIC.</p> <p>Navigation Path: Test &gt;Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFig &gt;IP &gt;Default Gateway&gt;Mode</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:MODE &lt;wsp&gt;AUTOMATIC   STATic</p>
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AUTOMATIC   STATic</p> <p>Selects the Default Gateway IPv6 Address mode.</p> <p>AUTOMATIC: Automatic</p> <p>STATic: Static</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:DGAT:IPV:MODE AUT</p> <p>SOUR:DATA:TEL:ETH:NETW:DGAT:IPV:MODE?</p> <p>Returns: AUTOMATIC</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:DGATeway:IPV:MODE</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:MODE?**

<b>Description</b>	<p>This query returns the Default Gateway IPv6 Address Mode.</p> <p>At *RST condition, this value is set to AUTOamtic.</p> <p>Navigation Path: Test &gt;Test Configurator&gt; Test Configurator&gt;Network Presence &gt; IPv6 &gt; CONFIg &gt;Default Gateway&gt;Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:NETWork:DGATeway:IPVersion:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Default Gateway IPv6 Address Mode.</p> <p>AUTOMATIC, Automatic is selected as default gateway address mode.</p> <p>MANUAL, Manual is selected as default gateway address mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:NETW:DGAT:IPV:MODE AUT</p> <p>SOUR:DATA:TEL:ETH:NETW:DGAT:IPV:MODE?</p> <p>Returns: AUTOMATIC</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:DGATeway:IPV:ADDRess?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:NETW:DGATeway:IPV:MODE</p>

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCal:IPVersion:ADDRESS**

<b>Description</b>	<p>This command sets the Link-Local IPv6 Address.</p> <p>At *RST condition, this value is set to ""FE80:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; Source Link-Local IPv6 Address.</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCal:IPVersion:ADDRESS &lt;wsp&gt; &lt;Stream&gt;, &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Link-Local IPv6 Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:LOC:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:LOC:IPV:ADDR 1, "FE80:0000:0000:0000:0000:0000:0000:FFFF"</p> <p>SOUR:DATA:TEL:ETH:PORT:LOC:IPV:ADDR? 1</p> <p>Returns: "FE80:0000:0000:0000:0000:0000:0000:FFFF"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPV:IICoupled</p>



**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCal:IPVersion:ADDRess?**

<b>Description</b>	<p>This query returns the Link-Local IPv6 Address.</p> <p>At *RST condition, this value is set to ""FE80:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; Source Link-Local IPv6 Address.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAl:IPVersion:ADDRess? <wsp><Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Link-Local IPv6 Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:LOC:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:LOC:IPV:ADDR 1, "FE80:0000:0000:0000:0000:0000:0000:FFFF"</p> <p>SOUR:DATA:TEL:ETH:PORT:LOC:IPV:ADDR? 1</p> <p>Returns: "FE80:0000:0000:0000:0000:0000:0000:FFFF"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPV:ADDRess

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCal:IPVersion:MODE**

<b>Description</b>	<p>This command selects the Link-Local IPv6 Mode.</p> <p>At *RST condition, this value is set to STATIC.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Link-Local IPv6 Address&gt;Mode</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCal:IPVersion:MODE &lt;wsp&gt; &lt;Stream&gt;, STATIC   SAUTO</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Mode:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: STATIC   SAUTO</p> <p>Selects the Link-Local IPv6 mode.</p> <p>STATIC: Static</p> <p>SAUTO: Stateless Auto.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:LOC:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:LOC:IPV:MODE? 1</p> <p>Returns: STAT</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPV:PMASK?</p>

<b>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAL:IPVersion:MODE?</b>	
<b>Description</b>	<p>This query returns the Link-Local IPv6 Mode.</p> <p>At *RST condition, this value is set to STATIC.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFIg &gt;IP&gt;Link-Local IPv6 Address&gt;Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAL:IPVersion:MODE? <wsp> <Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Link-Local IPv6 mode.</p> <p>STATic, Static is selected as Link-Local IPv6 mode.</p> <p>SAUTo, Stateless Auto. (SAUTo) is selected as Link-Local IPv6 mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:LOC:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:LOC:IPV:MODE? 1</p> <p>Returns: STAT</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAL:IPV:PMASK

## SCPI Command Reference

### IPv6 Address Configuration

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LOCAl:IPVersion:ADDRess:STATus?**

<b>Description</b>	<p>This query returns the Link-Local IPv6 address status.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP&gt;Link-Local IPv6 Address</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LOCAl:IPVersion:ADDRess:STATus? &lt;wsp&gt; &lt;Stream&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Link-Local IPv6 address status.</p> <p>TENTATIVE, Tentative is retrieved.</p> <p>GENERATING, Generating is retrieved.</p> <p>SUCCESSFUL, Successful is retrieved.</p> <p>PREFERRED, Preferred is retrieved.</p> <p>FAILED, Failed is retrieved.</p> <p>CHECKING, Checking is retrieved.</p> <p>NDUPLICATE, No Duplication is retrieved.</p> <p>DDETECTED, Duplication Detected is retrieved.</p> <p>UNDEFINED, Undefined is retrieved.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:STR:LOC:IPV:ADDR:STAT? 1</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBAl:IPV:STATus?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:ADDRESS**

<b>Description</b>	<p>This command sets the Global IPv6 Address.</p> <p>At *RST condition, this value is set to ""2001:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt;Global IPv6 Address.</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:ADDRESS &lt;wsp&gt; &lt;Stream&gt;, &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Global IPv6 Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:IIC 1, OFF</p> <p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:ADDR 1, "2001:0000:0000:0000:0000:0000:0000:0000"</p> <p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:ADDR? 1</p> <p>Returns: "2001:0000:0000:0000:0000:0000:0000:0000"</p>
<b>Note(s)</b>	<p>When Interface ID Coupled is enabled, Global IPV6 Address will configure up to 64 MSB and when it is disabled, it will configure up to 128 bits.</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion?</p>

---

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:ADDRess?**

<b>Description</b>	<p>This query returns the Global IPv6 Address.</p> <p>At *RST condition, this value is set to ""2001:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt;Global IPv6 Address.</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:ADDRess? &lt;wsp&gt; &lt;Stream&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Address&gt;</p>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Global IPv6 Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:MODE 1, STAT SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:IIC 1, OFF SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:ADDR 1, "2001:0000:0000:0000:0000:0000:0000:0000" SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:ADDR? 1 Returns: "2001:0000:0000:0000:0000:0000:0000:0000"</p>
<b>Note(s)</b>	<p>When Interface ID Coupled is enabled, Global IPV6 Address will configure up to 64 MSB and when it is disabled, it will configure up to 128 bits.</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:IPVersion</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:MODE**

<b>Description</b>	<p>This command selects the Global IPv6 Mode.</p> <p>At *RST condition, this value is set to NONE.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Global IPv6 Address &gt; Mode.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:MODE <wsp> <Stream>, NONE   STATIC   SAUTO
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Mode:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NONE   STATIC   SAUTO</p> <p>Selects the Global IPv6 mode.</p> <p>NONE, None</p> <p>STATIC: Static</p> <p>SAUTO: Stateless Auto.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:MODE? 1</p> <p>Returns: STATIC</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCal:IPV:ADDRESS?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCal:IPV:MODE</p>

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:MODE?**

<b>Description</b>	<p>This query returns the Global IPv6 Mode.</p> <p>At *RST condition, this value is set to NONE.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Global IPv6 Address &gt; Mode.</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:MODE? &lt;wsp&gt; &lt;Stream&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Mode&gt;</p>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Global IPv6 mode.</p> <p>NONE, No Global IPv6 mode is selected.</p> <p>STATIC, Static is selected as Global IPv6 mode.</p> <p>SAUTO, Stateless Auto. (SAUTo) is selected as Global IPv6 mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:MODE? 1</p> <p>Returns: STATIC</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAl:IPV:ADDRESS</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAl:IPV:MODE</p>



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:PMASK**

---

<b>Description</b>	<p>This command sets the Global IPv6 Address Prefix Mask.</p> <p>At *RST condition, this value is set to ""0000:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Global IPv6 Address &gt; Prefix Mask</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt; Services &gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Global IPv6 Address &gt; Prefix Mask</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:PMASK <wsp> <Stream>, <Prefix Mask>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Prefix Mask:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Global IPv6 Address Prefix Mask.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:PMAS 1, "FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000"</p> <p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:PMAS? 1</p> <p>Returns: "FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCal:IPV:MODE?

---

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:PMASK?**

<b>Description</b>	<p>This query returns the Global IPv6 Address Prefix Mask.</p> <p>At *RST condition, this value is set to ""0000:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Global IPv6 Address &gt; Prefix Mask</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:PMASK? &lt;wsp&gt; &lt;Stream&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Address&gt;</p>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Global IPv6 Address Prefix Mask.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:MODE 1, STAT SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:PMAS 1, "FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000" SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:PMAS? 1 Returns: "FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAl:IPV:MODE</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:IICoupled**

<b>Description</b>	<p>This command enables or disables the Interface ID of the Global IPv6 address.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Global IPv6 Address &gt;Interface ID Coupled.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:IICoupled <wsp> <Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables the Interface ID of the Global IPv6 address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:IIC 1, OFF</p> <p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:IIC? 1</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPV:IICoupled

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:IICoupled?**

<b>Description</b>	<p>This query returns the status of Interface ID of the Global IPv6 address.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Global IPv6 Address &gt;Interface ID Coupled.</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPVersion:IICoupled? &lt;wsp&gt; &lt;Stream&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Interface ID of the Global IPv6 address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:MODE 1, STAT SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:IIC 1, OFF SOUR:DATA:TEL:ETH:PORT:GLOB:IPV:IIC? 1 Returns: 0</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:ADDRESS SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:MODE SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:GLOB:IPV:IICoupled</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:IPVersion:ADDRess:STATus?**

<b>Description</b>	<p>This query returns the Global IPv6 address status.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Global IPv6 Address &gt; Global IPv6 Address</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:GLOBal:IPVersion:ADDRess:STATus? &lt;wsp&gt; &lt;Stream&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Global IPv6 address status.</p> <p>TENTATIVE, Tentative is retrieved.</p> <p>GENERATING, Generating is retrieved.</p> <p>SUCCESSFUL, Successful is retrieved.</p> <p>PREFERRED, Preferred is retrieved.</p> <p>FAILED, Failed is retrieved.</p> <p>CHECKING, Checking is retrieved.</p> <p>NDUPLICATE, No Duplication is retrieved.</p> <p>DDETECTED, Duplication Detected is retrieved.</p> <p>UNDEFINED, Undefined is retrieved.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:STR:GLOB:IPV:ADDR:STAT? 1</p>
<b>See Also</b>	<p>FETCH:DATA:TELEcom:ETHernet:PORT:LOCAl:IPV:STATus?</p>

## SCPI Command Reference

### IPv6 Address Configuration

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:ADDRESS**

<b>Description</b>	<p>This command sets the IPv6 Default Gateway Address.</p> <p>At *RST condition, this value is set to ""FE80:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Default Gateway&gt;Address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:ADDRESS &lt;wsp&gt; &lt;Stream&gt;, &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Default Gateway IPv6 Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:DGAT:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:DGAT:IPV:ADDR 1, "FE80:0000:0000:0000:0000:0000:0000:0000"</p> <p>SOUR:DATA:TEL:ETH:PORT:DGAT:IPV:ADDR? 1</p> <p>Returns: "FE80:0000:0000:0000:0000:0000:0000:0000"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPV:ADDRESS</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPV:MODE</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:ADDRess?**

<b>Description</b>	<p>This query returns the IPv6 Default Gateway Address.</p> <p>At *RST condition, this value is set to ""FE80:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Default Gateway&gt;Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:ADDRess? <wsp> <Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Default Gateway IPv6 Address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:DGAT:IPV:MODE 1, STAT</p> <p>SOUR:DATA:TEL:ETH:PORT:DGAT:IPV:ADDR 1, "FE80:0000:0000:0000:0000:0000:0000:0000"</p> <p>SOUR:DATA:TEL:ETH:PORT:DGAT:IPV:ADDR? 1</p> <p>Returns: "FE80:0000:0000:0000:0000:0000:0000:0000"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPV:MODE?

---

## SCPI Command Reference

### IPv6 Address Configuration

---

<b>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:MODE</b>	
<b>Description</b>	<p>This command selects the Default Gateway IPv6 Address Mode.</p> <p>At *RST condition, this value is set to AUTomatic.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Default Gateway&gt;Mode</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:MODE &lt;wsp&gt; &lt;Stream&gt;, AUTomatic   STATic</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Mode:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AUTomatic   STATic</p> <p>Selects the Default Gateway IPv6 Address mode.</p> <p>AUTomatic: Automatic</p> <p>STATic: Static</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:DGAT:IPV:MODE 1, AUT</p> <p>SOUR:DATA:TEL:ETH:PORT:DGAT:IPV:MODE? 1</p> <p>Returns: AUTOMATIC</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPV:MODE</p>



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:MODE?**

<b>Description</b>	<p>This query returns the Default Gateway IPv6 Address Mode.</p> <p>At *RST condition, this value is set to AUTomatic.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Default Gateway&gt;Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPVersion:MODE? <wsp> <Stream>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Default Gateway IPv6 Address Mode.</p> <p>AUTOMATIC, Automatic is selected as default gateway address mode.</p> <p>MANUAL, Manual is selected as default gateway address mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:DGAT:IPV:MODE 1, AUT</p> <p>SOUR:DATA:TEL:ETH:PORT:DGAT:IPV:MODE? 1</p> <p>Returns: AUTOMATIC</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPV:ADDRes?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:PORT:DGATeway:IPV:MODE</p>

---

## SCPI Command Reference

### IPv6 Address Configuration

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:DGATeway:IPVersion:ADDRess:STATus?**

<b>Description</b>	<p>This query returns the Default Gateway IPv6 address Status.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Test Configurator&gt; Set Up &gt;Stream&gt;MAC/IP/UDP &gt; IPv6 &gt; CONFig &gt;IP &gt;Default Gateway.</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:DGATeway:IPVersion:ADDRess:STATus? &lt;wsp&gt; &lt;Stream&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>EtherSAM - Selects the service from 1 to 10.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Default Gateway IPv6 address status.</p> <p>UNDEFINED, Undefined is retrieved.</p> <p>CHECKING, Checking is retrieved.</p> <p>UNREACHABLE, Unreachable is retrieved.</p> <p>REACHABLE, Reachable is retrieved.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:STR:DGAT:IPV:ADDR:STAT? 1</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:PORT:LOCAl:IPV:STATus?</p>

## RFC 6349

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE**

---

<b>Description</b>	<p>This command selects the operation mode for RFC 6349 test application.</p> <p>At *RST, this value is set to Dual Test Set.</p> <p>Navigation Path: RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Operation mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE <wsp>DTS
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DTS</p> <p>Selects the operation mode</p> <p>DTS: Dual test set</p> <p>EXFOWORX: EXFO   Worx Interop</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:OPER:MOD DTS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:DIRection

---

## SCPI Command Reference

RFC 6349

---

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE?

---

<b>Description</b>	<p>This query returns the operation mode for RFC 6349 test application.</p> <p>At *RST, this value is set to Dual Test Set.</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Operation mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Operation mode</p> <p>DTS: Dual test set</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:OPER:MOD DTS</p> <p>SOUR:DATA:TEL:ETH:RFC:OPER:MOD?</p> <p>Returns DTS</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:DIRection?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:DIRection**

<b>Description</b>	<p>This command selects the direction for RFC 6349 test application</p> <p>At *RST, this value is set to BIDIRECTIONAL.</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Direction</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:DIRection &lt;wsp&gt;BIDIRECTIONAL   LTOR   RTOL</code>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIDIRECTIONAL   LTOR   RTOL</p> <p>Selects the direction</p> <p>BIDIRECTIONAL: Bidirectional</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:RFC:DIR LTOR</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort</code>

---

## SCPI Command Reference

RFC 6349

---

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:DIRection?

---

<b>Description</b>	<p>This query returns the direction for RFC 6349 test application.</p> <p>At *RST, this value is set to BIDIRECTIONAL.</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Direction</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:DIRection?
<b>Response Syntax</b>	<Direction>
<b>Response(s)</b>	<p><b>Direction:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the direction</p> <p>BIDIRECTIONAL: Bidirectional</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:DIR LTOR</p> <p>SOUR:DATA:TEL:ETH:RFC:DIR?</p> <p>Returns LTOR</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort**

<b>Description</b>	<p>This command sets the TCP server port for RFC 6349 test application.</p> <p>At *RST, this value is set to 50201.</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; TCP Server port</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort &lt;wsp&gt;LTOR   RTOL, &lt;TCP server port&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the server port per direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>TCP server port:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the TCP server port value</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:RFC:TCP:SERP 24</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE</code>

---

## SCPI Command Reference

RFC 6349

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort?**

<b>Description</b>	<p>This query returns the direction for RFC 6349 test application.</p> <p>At *RST, this value is set to 50201</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; TCP Server port</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort? &lt;wsp&gt;LTOR   RTOL[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR and RTOL as direction, for Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p>
<b>Response Syntax</b>	<p>&lt;TCP server port&gt;</p>
<b>Response(s)</b>	<p><b>TCP server port:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the TCP server port value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:TCP:SERP 24</p> <p>SOUR:DATA:TEL:ETH:RFC:TCP:SERP?</p> <p>Returns 24</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE?</p>



**:SOURce[1..n]:DATA:TELecom:ETHernet:RFC:WSIZetarget**

<b>Description</b>	<p>This command selects the direction for RFC 6349 test application</p> <p>At *RST, this value is set to BIDIRECTIONAL.</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Direction</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELecom:ETHernet:RFC:WSIZetarget &lt;wsp&gt;WS1M   WS4M   WS8M   WS16M   WS32M</code>
<b>Parameter(s)</b>	<p><b>Value:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: WS1M   WS4M   WS8M   WS16M   WS32M</p> <p>Sets the Window size target</p> <p>WS1M: window size 1MiB</p> <p>WS4M: window size 4 MiB</p> <p>WS8M: window size 8 MiB</p> <p>WS16M: window size 16 MiB</p> <p>WS32M: window size 32 MiB</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:RFC:RFC:WSIZetarget</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELecom:ETHernet:RFC:TCP:SERPort</code>

---

## SCPI Command Reference

RFC 6349

---

---

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:WSIZetarget?**

---

<b>Description</b>	<p>This query returns the direction for RFC 6349 test application.</p> <p>At *RST, this value is set to BIDIRECTIONAL.</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Direction</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:WSIZetarget?
<b>Response Syntax</b>	<Window Size>
<b>Response(s)</b>	<p><b>Window Size:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Window size target</p> <p>WS1M: window size 1MiB</p> <p>WS4M: window size 4 MiB</p> <p>WS8M: window size 8 MiB</p> <p>WS16M: window size 16 MiB</p> <p>WS32M: window size 32 MiB</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:RFC:WSIZetarget WS4M</p> <p>SOUR:DATA:TEL:ETH:RFC:RFC:WSIZetarget?</p> <p>Returns WindowSize4MiB</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:SERPort?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:CIR**

<b>Description</b>	<p>This command sets the CIR for RFC 6349 test application.</p> <p>At *RST, this value is set to 1.0</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; CIR</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:CIR &lt;wsp&gt;LTOR   RTOL, &lt;CIR&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>CIR:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the CIR</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:CIR 2.2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TOSDs</p>

---

## SCPI Command Reference

RFC 6349

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:CIR?**

<b>Description</b>	<p>This query returns the CIR for RFC 6349 test application.</p> <p>At *RST, this value is set to 1.0</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; CIR</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:CIR? &lt;wsp&gt;LTOR   RTOL[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR and RTOL as direction, for Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p>
<b>Response Syntax</b>	<p>&lt;CIR&gt;</p>
<b>Response(s)</b>	<p><b>CIR:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the CIR value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:CIR 2.2</p> <p>SOUR:DATA:TEL:ETH:RFC:CIR? Returns 2.2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TOSDs?</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TOSDs**

<b>Description</b>	<p>This command sets the TOS/DS for RFC 6349 test application.</p> <p>At *RST, this value is set to 0</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; TOS/DS</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TOSDs <wsp>LTOR   RTOL, <TOSDS>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>TOSDS:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the TOS/DS value</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:TOSD LTOR,#H01
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:CIR

---

## SCPI Command Reference

RFC 6349

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TOSDs?**

<b>Description</b>	<p>This query returns the TOS/DS for RFC 6349 test application.</p> <p>At *RST, this value is set to 0</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; TOS/DS</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TOSDs? &lt;wsp&gt;LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR and RTOL as direction, for Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;TOS/DS&gt;</p>
<b>Response(s)</b>	<p><b>TOS/DS:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the TOS/DS value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:TOSD LTOR,#H01</p> <p>SOUR:DATA:TEL:ETH:RFC:TOSD? LTOR Returns 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:CIR?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:MULTiple:CONNECTIONs**

---

<b>Description</b>	<p>This command enable disables Multiple connections</p> <p>At *RST, this value is set to 1</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Multiple Connections</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:MULTiple:CONNECTIONs <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable/disable the Multiple connections</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:MULT:CONN ON
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE

---

## SCPI Command Reference

RFC 6349

---

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:MULTiple:CONNECTIONs?

---

<b>Description</b>	This query returns status of Multiple connection At *RST, this value is set to 1 Navigation Path: Setup > RFC 6349 > Test Configurator > RFC 6349 > Multiple Connections
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:MULTiple:CONNECTIONs?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Multiple connection status 1, Multiple connection is enabled. 0, Multiple connection is disabled.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:MULT:CONN ON SOUR:DATA:TEL:ETH:RFC:MULT:CONN? Returns 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:OPERation:MODE?

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:MAX:MTU**

---

<b>Description</b>	<p>This command sets the Max MTU value</p> <p>At *RST, this value is set to 1500</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Max MTU</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:MAX:MTU &lt;wsp&gt; &lt;Value&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Value:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Max MTU value.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:RFC:MAX:MTU 1080</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:PATH:MTU:DISCcovery</code>

---

## SCPI Command Reference

RFC 6349

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:MAX:MTU?**

<b>Description</b>	This query returns the Max MTU for RFC 6349 test application. At *RST, this value is set to 1500 Navigation Path: Setup > RFC 6349 > Test Configurator > RFC 6349 > Max MTU
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:MAX:MTU?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are: MAXimum   MINimum
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Max MTU value.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:MAX:MTU 1080 SOUR:DATA:TEL:ETH:RFC:MAX:MTU? Returns 1080
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:PATH:MTU:DISCcovery?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:PATH:MTU:DISCoverY**

<b>Description</b>	<p>This command sets the status for Path MTU discovery.</p> <p>At *RST, this value is set to 0</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Path MTU Discovery</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:PATH:MTU:DISCoverY <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable/disable the Path MTU Discovery.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:PATH:MTU:DISC ON
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:MAX:MTU

---

## SCPI Command Reference

RFC 6349

---

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:PATH:MTU:DISCovey?

---

<b>Description</b>	This query returns the status for Path MTU discovery. At *RST, this value is set to 0 Navigation Path: Setup > RFC 6349 > Test Configurator > RFC 6349 > Path MTU Discovery
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:PATH:MTU:DISCovey?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Path MTU Discovery status. 1, Path MTU Discovery is enabled. 0, Path MTU Discovery is disabled.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:PATH:MTU:DISC ON SOUR:DATA:TEL:ETH:RFC:PATH:MTU:DISC? Returns 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:MAX:MTU?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:DURation**

<b>Description</b>	<p>This command sets the TCP Throughput duration.</p> <p>At *RST, this value is set to ""00d:00:01:00""</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; TCP Throughput &gt; Duration</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:DURation &lt;wsp&gt; &lt;Duration&gt;</code>
<b>Parameter(s)</b>	<p><b>Duration:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the TCP throughput duration.</p> <p>Format is ""00d:hh:mm:ss""</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETHernet:RFC:TCP:THR:DUR "00d:00:01:01"</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold</code>

---

## SCPI Command Reference

RFC 6349

---

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:DURation?

---

<b>Description</b>	<p>This query returns the TCP Throughput duration.</p> <p>At *RST, this value is set to ""00d:00:01:00""</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; TCP Throughput &gt; Duration</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:DURation?
<b>Response Syntax</b>	<Duration>
<b>Response(s)</b>	<p><b>Duration:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the TCP Throughput duration</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETHernet:RFC:TCP:THR:DUR "00d:00:01:01"</p> <p>SOUR:DATA:TEL:ETHernet:RFC:TCP:THR:DUR? Returns "00d:00:01:01"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold**

<b>Description</b>	<p>This command sets the TCP Throughput Threshold.</p> <p>At *RST, this value is set to 95.0</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; TCP Throughput &gt; Threshold(%)</p> <p>Navigation Path: Results &gt; Summary &gt; TCP Throughput &gt; TCP Throughput Threshold</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold &lt;wsp&gt; &lt;Threshold&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Threshold:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the TCP Throughput Threshold.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:RFC:TCP:THR:THR 20.2</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:DURation</code>

---

## SCPI Command Reference

RFC 6349

---

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold?

---

<b>Description</b>	<p>This query returns the TCP Throughput Threshold.</p> <p>At *RST, this value is set to 95.0</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; TCP Throughput &gt; Threshold(%)</p> <p>Navigation Path: Results &gt; Summary &gt; TCP Throughput &gt; TCP Throughput Threshold</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold? &lt;wsp&gt;MAXimum   MINimum</p>
<b>Response Syntax</b>	<p>&lt;Threshold&gt;</p>
<b>Response(s)</b>	<p><b>Threshold:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Threshold value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:TCP:THR:THR 20.2</p> <p>SOUR:DATA:TEL:ETH:RFC:TCP:THR:THR? Returns 20.2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:DURation?</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:VERDict**

---

<b>Description</b>	<p>This command enable disables the TCP Throughput Verdict</p> <p>At *RST, this value is set to 1</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; TCP Throughput &gt; Verdict</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:VERDict <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable/disable the TCP throughput verdict.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:TCP:THR:VERD 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold

---

## SCPI Command Reference

RFC 6349

---

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:VERDict?

---

<b>Description</b>	<p>This query returns status of the TCP Throughput Verdict</p> <p>At *RST, this value is set to 1</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; TCP Throughput &gt; Verdict</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:VERDict?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the verdict status</p> <p>1, verdict is enabled.</p> <p>0, verdict is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:TCP:THR:VERD 1</p> <p>SOUR:DATA:TEL:ETH:RFC:TCP:THR:VERD? Returns 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep**

---

<b>Description</b>	<p>This command sets the Window sweep status</p> <p>At *RST, this value is set to 0</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Window Sweep</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable/disable the Window sweep</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:RFC:WIND:SWE ON</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:VERDict</code>

---

## SCPI Command Reference

RFC 6349

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep?**

<b>Description</b>	This query returns the Window sweep status. At *RST, this value is set to 0 Navigation Path: Setup > RFC 6349 > Test Configurator > RFC 6349 > Window Sweep
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Window sweep status 1, Window sweep is enabled. 0, Window sweep is disabled.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:WIND:SWE ON SOUR:DATA:TEL:ETH:RFC:WIND:SWE? Returns 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:VERDict?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:DURation**

<b>Description</b>	<p>This command sets the Window sweep status.</p> <p>At *RST, this value is set to ""00:30"</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Window Sweep &gt; Duration (per step)</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:DURation &lt;wsp&gt; &lt;Duratoin&gt;</pre>
<b>Parameter(s)</b>	<p><b>Duratoin:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Window sweep duration.</p> <p>Format is ""MM:SS""</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:RFC:WIND:SWE:DUR "00:30"</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep</pre>

---

## SCPI Command Reference

RFC 6349

---

---

### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:DURation?

---

<b>Description</b>	<p>This query returns the Window sweep status.</p> <p>At *RST, this value is set to ""00:30""</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Window Sweep &gt; Duration (per step)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:DURation?
<b>Response Syntax</b>	<Duration>
<b>Response(s)</b>	<p><b>Duration:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Window sweep duration</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:WIND:SWE:DUR "00:30"</p> <p>SOUR:DATA:TEL:ETH:RFC:WIND:SWE:DUR? Returns "00:30"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:RESTore:DEFault**

---

<b>Description</b>	<p>This command restore defaults the RFC 6349 values.</p> <p>This is a action commands and has no default value.</p> <p>Navigation Path: Setup &gt; RFC 6349 &gt; Test Configurator &gt; RFC 6349 &gt; Restore RFC 6349 Defaults.</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:RESTore:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:REST:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:RESTore:DEFault

---

## Modify Trib Slots/Channels (Multi-Channel OTN)

---

:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries

---

<b>Description</b>	<p>This command sets ODUx Tributaries configuration (Port Id/ Slot Id Associations) for the given direction to the user specified values</p> <p>N.B. Partial list of ODUx Tributary configuration is allowed and will be auto-completed by assigning free increasing tributary slot id to free increasing tributary port id.</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4 lanes)&gt;ODU Channels(ODUx) &gt; Modify Tributary Slots/Channels</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries &lt;wsp&gt; &lt;OduType&gt;, TX   RX   RXANDTX, &lt;List of ODUx Tributary Configuration</p>

---



:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries

**Parameter(s)**

**OduType:**

The program data syntax for the first parameter is defined as a <STRING PROGRAM DATA> element.

Selects the Parent ODU type value.

Example: Type ODU4 when mapping is ODU4/ODU0 and we want to configure ODU0 Tributaries (Slot Id/Port Id)

**Direction:**

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: TX | RX | RXANDTX

Selects the channel direction to apply the tributaries configuration

TX: TX valid only when the Coupled Status is OFF, invalid otherwise

RX: RX valid only when the Coupled Status is OFF, invalid otherwise

RXANDTX: RXANDTX valid only when the Coupled Status is ON, invalid otherwise

**List of ODUx Tributary Configuration**

The program data syntax for the third parameter is defined as a <STRING PROGRAM DATA> element.

The allowed elements for this parameter are: Complete or Partial List of ODUx Tributary configuration for specified parent ODU type.

**Example(s)**

SOUR:DATA:TEL:OTN:TRIB  
ODU4,rxandtx,"(ODU2,1,(80,79,78,77,76,75,74,73),ODU2,10,(33:34,35,36,37:40))" when  
SOUR:DATA:TEL:OTN:TRIB:COUP? = 1

SOUR:DATA:TEL:OTN:TRIB  
ODU4,rx,"(ODU2,1,(80,79,78,77,76,75,74,73),ODU2,10,(33,34,35,36:37,38,39,40))" when  
SOUR:DATA:TEL:OTN:TRIB:COUP? = 0

SOUR:DATA:TEL:OTN:TRIB  
ODU4,tx,"(ODU2,1,(80,79,78:73),ODU2,10,(33,34,35,36,37,38,39,40))" when  
SOUR:DATA:TEL:OTN:TRIB:COUP? = 0

## SCPI Command Reference

### *Modify Trib Slots/Channels (Multi-Channel OTN)*

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries**

**See Also**

- SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries?
- SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled
- SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled?
- SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:DEFAult
- SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COPI

---

**:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries?**

<b>Description</b>	<p>This query returns the complete list of ODUx Tributaries configuration (Port Id/ Slot Id Associations) for the given direction</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4 lanes)&gt;ODU Channels(ODUx) &gt; Modify Tributary Slots/Channels</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries? <wsp> <OduType>, TX   RX   RXANDTX
<b>Parameter(s)</b>	<p><b>OduType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Parent ODU type value.</p> <p>Example: Type ODU4 when mapping is ODU4/ODU0 and we want to configure ODU0 Tributaries (Slot Id/Port Id)</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX   RX   RXANDTX</p> <p>Selects the channel direction to apply the tributaries configuration</p> <p>TX: TX valid only when the Coupled Status is OFF, invalid otherwise</p> <p>RX: RX valid only when the Coupled Status is OFF, invalid otherwise</p> <p>RXANDTX: RXANDTX valid only when the Coupled Status is ON, invalid otherwise</p>
<b>Response Syntax</b>	<List of ODUx Tributary Configuration

## SCPI Command Reference

### Modify Trib Slots/Channels (Multi-Channel OTN)

---

:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries?

#### Response(s)

#### List of ODUx Tributary Configuration

The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element.

Returns all ODUx Tributary configuration for specified parent ODU type.

Ex: Using multi-channel test application with ODU4/ODU2 mapping and after configuring the tributaries at default the command TRIB? returns

```
""(ODU2,1,(1,2,3,4,5,6,7,8),ODU2,2,(9,10,11,12,13,14,15,16),ODU2,3,(17,18,19,20,21,22,23,24),ODU2,4,(25,26,27,28,29,30,31,32),ODU2,5,(33,34,35,36,37,38,39,40),ODU2,6,(41,42,43,44,45,46,47,48),ODU2,7,(49,50,51,52,53,54,55,56),ODU2,8,(57,58,59,60,61,62,63,64),ODU2,9,(65,66,67,68,69,70,71,72),ODU2,10,(73,74,75,76,77,78,79,80))""
```

#### Example(s)

SOUR:DATA:TEL:OTN:TRIB? ODU4,rxandtx when

SOUR:DATA:TEL:OTN:TRIB:COUP? = 1

SOUR:DATA:TEL:OTN:TRIB? ODU4,tx when SOUR:DATA:TEL:OTN:TRIB:COUP? = 0

SOUR:DATA:TEL:OTN:TRIB? ODU4,rx when SOUR:DATA:TEL:OTN:TRIB:COUP? = 0

#### See Also

SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries

SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled

SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled?

SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:DEFAult

SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COPI

**:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:DEFAult**

<b>Description</b>	<p>This command sets ODUx Tributaries configuration (Port Id/ Slot Id Associations) for the given direction to the default values</p> <p>N.B. Default ODUx Tributary configuration is a group of increasing tributary slot id assigned to increasing tributary port id.</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4 lanes)&gt;ODU Channels(ODUx) &gt; Modify Tributary Slots/Channels</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:DEFAult <wsp> <OduType>, TX   RX   RXANDTX
<b>Parameter(s)</b>	<p><b>OduType:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Parent ODU type value.</p> <p>Example: Type ODU4 when mapping is ODU4/ODU0 and we want to configure ODU0 Tributaries (Slot Id/Port Id)</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX   RX   RXANDTX</p> <p>Selects the channel direction to apply the tributaries configuration</p> <p>TX: TX valid only when the Coupled Status is OFF, invalid otherwise</p> <p>RX: RX valid only when the Coupled Status is OFF, invalid otherwise</p> <p>RXANDTX: RXANDTX valid only when the Coupled Status is ON, invalid otherwise</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:TRIB:DEFA RXANDTX when SOUR:DATA:TEL:OTN:TRIB:COUP? = 1</p> <p>SOUR:DATA:TEL:OTN:TRIB:DEFA RX when SOUR:DATA:TEL:OTN:TRIB:COUP? = 0</p> <p>SOUR:DATA:TEL:OTN:TRIB:DEFA TXwhen SOUR:DATA:TEL:OTN:TRIB:COUP? = 0</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries?</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled?</p>

## SCPI Command Reference

### *Modify Trib Slots/Channels (Multi-Channel OTN)*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COPI**

<b>Description</b>	Copy the received ODU <sub>x</sub> Tributaries configuration (Port Id/ Slot Id Associations) into the RX ODU <sub>x</sub> Tributaries configuration (Port Id/ Slot Id Associations) Navigation Path: OTN BERT Test > Test > Setup > OTU (4 lanes)>ODU Channels(ODU <sub>x</sub> ) > Modify Tributary Slots/Channels
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COPI
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:TRIB:COPI
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries? SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled?

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled**

<b>Description</b>	<p>This command sets the Coupled Status for the ODUx Tributaries Configuration</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4 lanes)&gt;ODU Channels(ODUx) &gt; Modify Tributary Slots/Channels</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled &lt;wsp&gt;ON   OFF[, TXTORX   RXTOTX]</p>
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Set the coupled status as specified of the ODU Tributaries configuration (Port Id/ Slot Id Associations)</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TXTORX   RXTOTX</p> <p>Selects the source to destination direction to copy the tributaries configuration</p> <p>TXTORX valid only when the Coupled Status is ON, will copy the TX tributaries configurarion over the RX tributaries configurarion.</p> <p>RXTOTX valid only when the Coupled Status is ON, will copy the RX tributaries configurarion over the TX tributaries configurarion.</p> <p>Invalid when the Coupled Status is OFF</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:TRIB:COUP ON TXTORX</p> <p>SOUR:DATA:TEL:OTN:TRIB:COUP ON RXTOTX</p> <p>SOUR:DATA:TEL:OTN:TRIB:COUP OFF</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled?</p> <p>SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries</p> <p>SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries?</p> <p>SOURCE[1..n]:DATA:TELEcom:OTN:TRIButaries:DEFAult</p>

## SCPI Command Reference

### *Modify Trib Slots/Channels (Multi-Channel OTN)*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled?**

<b>Description</b>	<p>This query returns the Coupled Status for the ODUx Tributaries Configuration</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: OTN BERT Test &gt; Test &gt; Setup &gt; OTU (4 lanes)&gt;ODU Channels(ODUx) &gt; Modify Tributary Slots/Channels</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled?
<b>Response Syntax</b>	<Coupled Status>
<b>Response(s)</b>	<p><b>Coupled Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the slot status, whether it is selected or not.</p> <p>1, returns the slot value as ON.</p> <p>0, returns the slot value as OFF.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:TRIB:COUP?
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:COUPled</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries?</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:TRIButaries:DEFAult</p>

---



# Timer

---

**:SOURce[1..n]:DATA:TELEcom:TIMer:CONFig**

---

<b>Description</b>	<p>This command automatically starts and/or stops a test case at a given time or for a specific duration.</p> <p>At *RST condition, this value is set to 15 minutes.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Duration ON/OFF</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Start Time ON/OFF</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Stop Time ON/OFF</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TIMer:CONFig <wsp>STARttime   STOPtime   DURation, ON   OFF
<b>Parameter(s)</b>	<p><b>Time:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: STARttime   STOPtime   DURation</p> <p>STARttime: the time the created test will automatically start.</p> <p>STOPtime: the time the test will automatically stop.</p> <p>DURation: the test duration based on the test delayed start time.</p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables manual test.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TIM:CONF STAR,ON</p> <p>SOUR:DATA:TEL:TIM:CONF? STAR</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TIMer:CONFig</p> <p>SOURce[1..n]:DATA:TELEcom:TIMer:STARt?</p>

---

## SCPI Command Reference

### Timer

---

:SOURce[1..n]:DATA:TELEcom:TIMer:CONFig?

<b>Description</b>	<p>This query returns the status of the test start time, stop time, or duration.</p> <p>At *RST condition, this value is set to 15 minutes.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Duration ON/OFF</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Start Time ON/OFF</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Stop Time ON/OFF</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TIMer:CONFig? <wsp>STARttime   STOptime   DURATION
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: STARttime   STOptime   DURATION</p> <p>STARttime, returns the time the created test will automatically start.</p> <p>STOptime, returns the time the test will automatically stop.</p> <p>DURATION, returns the test duration based on the test delayed start time.</p>
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of selected parameter (STARttime, STOptime, DURATION).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TIM:CONF STAR,ON</p> <p>SOUR:DATA:TEL:TIM:CONF? STAR</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TIMer:CONFig</p> <p>SOURce[1..n]:DATA:TELEcom:TIMer:STARt?</p>

---

**:SOURce[1..n]:DATA:TELEcom:TIMer:STARt**

---

<b>Description</b>	<p>This command allows you to select the specific date and time the created test will automatically start. Start time should be enabled.</p> <p>At *RST condition, this value is set to the current date.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Start Time &gt; Date</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:TIMer:STARt &lt;wsp&gt; &lt;DateTime&gt;</code>
<b>Parameter(s)</b>	<p><b>DateTime:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Select the specific date and time the test will automatically start.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TIM:STAR "7/15/2011 20:30:35" SOUR:DATA:TEL:TIM:STAR? Returns: "7/15/2011 20:30:35"</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:TIMer:CONFig?</code>

---

## SCPI Command Reference

### Timer

---

**:SOURce[1..n]:DATA:TELEcom:TIMer:STARt?**

<b>Description</b>	<p>This query returns the specific date and time the created test case will automatically start. Start time should be enabled.</p> <p>At *RST condition, this value is set to the current date and time.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Start Time &gt; Date</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TIMer:STARt?
<b>Response Syntax</b>	<DateTime>
<b>Response(s)</b>	<p><b>DateTime:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the specific date the created test case will automatically start.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TIM:STAR "7/15/2011 20:30:35"</p> <p>SOUR:DATA:TEL:TIM:STAR?</p> <p>Returns: "7/15/2011 20:30:35"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TIMer:CONFig

---

---

**:SOURce[1..n]:DATA:TELEcom:TIMer:STOP**

<b>Description</b>	<p>This command allows you to select the specific date and time the test will automatically stop. Stop time should be enabled.</p> <p>At *RST condition, this value is set to the current date and time.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Stop Time &gt; Date</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:TIMer:STOP &lt;wsp&gt; &lt;DateTime&gt;</code>
<b>Parameter(s)</b>	<p><b>DateTime:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Select the specific date the test will automatically stop.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TIM:STOP "7/15/2011 20:30:35" SOUR:DATA:TEL:TIM:STOP? Returns: "7/15/2011 20:30:35"</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:TIMer:STOP? SOURce[1..n]:DATA:TELEcom:TIMer:STARt?</pre>

---

## SCPI Command Reference

### Timer

---

**:SOURce[1..n]:DATA:TELEcom:TIMer:STOP?**

<b>Description</b>	<p>This query returns the specific date and time the test will automatically stop. Stop time should be enabled.</p> <p>At *RST condition, this value is set to the current date and time.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Stop Time &gt; Date</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TIMer:STOP?
<b>Response Syntax</b>	<DateTime>
<b>Response(s)</b>	<p><b>DateTime:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the specific date and time the test case will automatically stop.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TIM:STOP "7/15/2011 20:30:35"</p> <p>SOUR:DATA:TEL:TIM:STOP?</p> <p>Returns: "7/15/2011 20:30:35"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TIMer:STOP</p> <p>SOURce[1..n]:DATA:TELEcom:TIMer:START</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:TIMer**

<b>Description</b>	<p>This command allows you to enable/disable the test timer. Start time should be enabled and time should have been expired.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Arm</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TIMer <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Allows you to enable/disable the test timer.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TIM ON</p> <p>SOUR:DATA:TEL:TIM?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TIMer?</p> <p>SOURce[1..n]:DATA:TELEcom:TIMer:STOP</p>

---

## SCPI Command Reference

### Timer

---

**:SOURce[1..n]:DATA:TELEcom:TIMer?**

<b>Description</b>	<p>This query returns the status of the timer. Start time should be enabled and time should have been expired.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Arm</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TIMer?
<b>Response Syntax</b>	<STATUS>
<b>Response(s)</b>	<p><b>STATUS:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the timer.</p> <p>1, timer is enabled.</p> <p>0, timer is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TIM ON</p> <p>SOUR:DATA:TEL:TIM?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TIMer</p> <p>SOURce[1..n]:DATA:TELEcom:TIMer:STOP</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:TIMer:DURation**

---

<b>Description</b>	<p>This command allows to select the test duration based on the test start time.</p> <p>At *RST condition, this value is set to 15 minutes.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Duration ON</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Duration &gt; Duration</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:TIMer:DURation &lt;wsp&gt;15M   1H   2H   6H   12H   24H   72H   7D   UDEFined</code>
<b>Parameter(s)</b>	<p><b>TIME:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 15M   1H   2H   6H   12H   24H   72H   7D   UDEFined</p> <p>Allows to select the test duration based on the test delayed start time.</p> <p>15M, sets the test duration to 15 minutes.</p> <p>1H, sets the test duration to 1 hour.</p> <p>2H, sets the test duration to 2 hours.</p> <p>6H, sets the test duration to 6 hours.</p> <p>12H, sets the test duration to 12 hours.</p> <p>24H, sets the test duration to 24 hours.</p> <p>72H, sets the test duration to 72 hours.</p> <p>7D, sets the test duration to 7D.</p> <p>UDEFined, sets the test duration to undefined.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TIM:DUR 15M SOUR:DATA:TEL:TIM:DUR? Returns: 15M</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:TIMer:DURation?</code>

---

## SCPI Command Reference

### Timer

---

:SOURce[1..n]:DATA:TELEcom:TIMer:DURation?	
<b>Description</b>	<p>This query returns the test duration based on the test delayed start time.</p> <p>At *RST condition, this value is set to 15 minutes.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Duration ON</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Duration &gt; Duration</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TIMer:DURation?
<b>Response Syntax</b>	<TIME>
<b>Response(s)</b>	<p><b>TIME:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the test duration based on the test delayed start time.</p> <p>15M, test duration is 15 minutes.</p> <p>1H, test duration is 1 hour.</p> <p>2H, test duration is 2 hours.</p> <p>6H, test duration is 6 hours.</p> <p>12H, test duration is 12 hours.</p> <p>24H, test duration is 24 hours.</p> <p>72H, test duration is 72 hours.</p> <p>7D, test duration is 7D.</p> <p>UDEFined, the test duration to undefined.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TIM:DUR 15M</p> <p>SOUR:DATA:TEL:TIM:DUR?</p> <p>Returns: 15M</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TIMer:DURation

---

**:SOURce[1..n]:DATA:TELEcom:TIMer:UDEF**

<b>Description</b>	<p>This command allows you to select the test duration, when User Defined has been selected for duration. Choices are from 1 second to 30 days.</p> <p>At *RST condition, this value is set to 1 second.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Duration ON</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Duration &gt; User Defined Duration</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:TIMer:UDEF &lt;wsp&gt; &lt;Utime&gt;</code>
<b>Parameter(s)</b>	<p><b>Utime:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the test duration, when User Defined is selected for duration.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TIM:UDEF "00d:00:15:00" SOUR:DATA:TEL:TIM:UDEF? Returns: "00d:00:15:00"</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:TIMer:UDEF? SOURce[1..n]:DATA:TELEcom:TIMer:DURATION</pre>

---

## SCPI Command Reference

### Timer

---

:SOURce[1..n]:DATA:TELEcom:TIMer:UDEF?

<b>Description</b>	<p>This query returns the test duration, when User Defined has been selected for duration.</p> <p>At *RST condition, this value is set to 1 second.</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Duration ON</p> <p>Navigation Path: Test Setup &gt; Timer &gt; Duration &gt; User Defined Duration</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TIMer:UDEF?
<b>Response Syntax</b>	<User Defined Duration>
<b>Response(s)</b>	<p><b>User Defined Duration:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the test duration when User Defined has been selected for duration.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TIM:UDEF "00d:00:15:00"</p> <p>SOUR:DATA:TEL:TIM:UDEF?</p> <p>Returns: "00d:00:15:00"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TIMer:UDEFined</p> <p>SOURce[1..n]:DATA:TELEcom:TIMer:DURation</p>

---

## System

---

**:SOURce[1..n]:DATA:TELEcom:FACTory:RESTore:DEFault**

---

<b>Description</b>	This command restores the factory default settings. Navigation Path: Test Setup > System > Factory Defult > Restore Default.
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:FACTory:RESTore:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:FACT:REST:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:RESTore:DEFault

---

## Summary

---

**:FETCh[1..n]:DATA:TELEcom:TEST:STARt:TIME?**

---

<b>Description</b>	This query returns the time at which the test started. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Summary > StartTime
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:TEST:STARt:TIME?
<b>Response Syntax</b>	<StartTime>
<b>Response(s)</b>	<b>StartTime:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the time at which the test started.
<b>Example(s)</b>	SOUR:DATA:TEL:TEST ON FETC:DATA:TEL:TEST:STAR:TIME?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:LOGGer:LIST?

---

---

**:FETCh[1..n]:DATA:TELecom:TEST:STATus?**

---

<b>Description</b>	<p>This query returns the test status.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Results &gt; Summary &gt; Status</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:TEST:STATus?
<b>Response Syntax</b>	<Test Status >
<b>Response(s)</b>	<p><b>Test Status :</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the status of the test.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TEST ON</p> <p>FETC:DATA:TEL:TEST:STAT?</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:TEST:START:TIME?

---

## SCPI Command Reference

### Summary

---

---

#### :FETCh[1..n]:DATA:TELEcom:TEST:POWer:RECOvery:COUNT?

---

<b>Description</b>	This query returns the number of times the system has recovered from power failure. At *RST condition, this value is set to device-dependent. Navigation Path: Results > Summary > Power Failure
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:TEST:POWer:RECOvery:COUNT?
<b>Response Syntax</b>	<Power Failure Status>
<b>Response(s)</b>	<b>Power Failure Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the power failure status of the test.
<b>Example(s)</b>	SOUR:DATA:TEL:TEST ON FETC:DATA:TEL:TEST:POW:REC:COUN?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:TEST:STATus?

---



---

**:FETCh[1..n]:DATA:TELEcom:TEST:STATus:VERDict?**

---

<b>Description</b>	<p>This query returns Global Test Status verdict status.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test App &gt; Test Setup &gt; Results &gt; Summary &gt; Status</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:TEST:STATus:VERDict?
<b>Response Syntax</b>	<Global Verdict>
<b>Response(s)</b>	<p><b>Global Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns test status global verdict status</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	FETC:DATA:TEL:TEST:STAT:VERD?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:TEST:STATus?

---

## SCPI Command Reference

### Summary

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern**

<b>Description</b>	<p>This command enables or disables the status of the BERT alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Alarms</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the BERT alarm generation.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ALAR:PATT ON</p> <p>SOUR:DATA:TEL:PATT:ALAR:PATT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AMount</p> <p>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:INJect</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern?**

---

<b>Description</b>	<p>This query returns the status of BERT alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: BERT &gt; Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Alarms</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of BERT alarm generation.</p> <p>1, alarm generation is enabled.</p> <p>0, alarm generation is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ALAR:PATT ON</p> <p>SOUR:DATA:TEL:PATT:ALAR:PATT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:INJect</p>

---

## SCPI Command Reference

### Summary

---

#### :FETCh[1..n]:DATA:TELecom:PATtern:ALARm:SEConds?

<b>Description</b>	<p>This query returns the number of seconds within which pattern alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Unframed &gt; Results &gt; Unframed Summary (BER)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:PATtern:ALARm:SEConds? <wsp> <Lane>,   PLOSs
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted alarm pattern. The range for the lane is from 0 to 19.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   PLOSs  </p> <p>Selects the type of pattern alarm.</p> <p>PLOSs: Pattern Loss</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the number of seconds of Pattern alarm.</p> <p>PLOSs, returns the type of pattern alarm as Pattern Loss.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PATT:ALAR:SEC? 1, PLOS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:PATtern:ALARm:PATtern:SEConds?

---

**:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:HISTory?**

<b>Description</b>	<p>This query returns the history status of pattern alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Unframed &gt; Results &gt; Unframed Summary (BER)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:HISTory? <wsp> <Lane>,   PLOsS
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted alarm pattern. The range for the lane is from 0 to 19.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   PLOsS  </p> <p>Selects the type of pattern alarm.</p> <p>PLOsS: Pattern Loss</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of the Pattern alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PATT:ALAR:HIST? 1, PLOS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CURRent?

## SCPI Command Reference

### Summary

---

---

#### :FETCh[1..n]:DATA:TELecom:PATtern:ALARm:CURRent?

---

<b>Description</b>	<p>This query returns the current status of pattern alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Unframed &gt; Results &gt; Unframed Summary (BER)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:PATtern:ALARm:CURRent? <wsp> <Lane>,   PLOSs
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted alarm pattern. The range for the lane is from 0 to 19.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   PLOSs  </p> <p>Selects the type of pattern alarm.</p> <p>PLOSs: Pattern Loss</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of Pattern alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that there is no alarm.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PATT:ALAR:CURR? 1, PLOS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:PATtern:ALARm:PATtern:HISTory?

---

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the pattern error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Unframed &gt; Results &gt; Unframed Summary (BER)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:SEConds? <wsp> <Lane>, BIT   MISMATCH0   MISMATCH1
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted error pattern. The range for the lane is from 0 to 19.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT   MISMATCH0   MISMATCH1</p> <p>Selects the type of pattern error.</p> <p>BIT: Bit</p> <p>MISMATCH0: Mismatch '0'</p> <p>MISMATCH1: Mismatch '1'</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the number of seconds of Pattern error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PATT:ERR:SEC? 1, BIT
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:SEConds?

## SCPI Command Reference

### Summary

---

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:COUNT?**

<b>Description</b>	<p>This query returns the count of pattern error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Unframed &gt; Results &gt; Unframed Summary (BER)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:COUNT? &lt;wsp&gt; &lt;Lane&gt;, BIT   MISMATCH0   MISMATCH1</p>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted error pattern. The range for the lane is from 0 to 19.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT   MISMATCH0   MISMATCH1</p> <p>Selects the type of pattern error.</p> <p>BIT: Bit</p> <p>MISMATCH0: Mismatch '0'</p> <p>MISMATCH1: Mismatch '1'</p>
<b>Response Syntax</b>	<p>&lt;Count&gt;</p>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the count of Pattern error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:PATT:ERR:COUN? 1, BIT</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CURREnt?</p>



**:FETCh[1..n]:DATA:TELeom:PATtern:ERRor:RATE?**

<b>Description</b>	<p>This query returns the current rate of pattern error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Unframed &gt; Results &gt; Unframed Summary (BER)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:PATtern:ERRor:RATE? <wsp> <Lane>, BIT   MISMATCH0   MISMATCH1
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted error pattern. The range for the lane is from 0 to 19.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT   MISMATCH0   MISMATCH1</p> <p>Selects the type of pattern error.</p> <p>BIT: Bit</p> <p>MISMATCH0: Mismatch '0'</p> <p>MISMATCH1: Mismatch '1'</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the current rate of Pattern error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PATT:ERR:RATE? 1,BIT
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:PATtern:ERRor:PATtern:HISTory?

## SCPI Command Reference

### Summary

---

---

#### :FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:HISTory?

---

<b>Description</b>	<p>This query returns the history status of pattern error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Unframed &gt; Results &gt; Unframed Summary (BER)</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:HISTory? &lt;wsp&gt; &lt;Lane&gt;, BIT   MISMATCH0   MISMATCH1</code>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted error pattern. The range for the lane is from 0 to 19.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT   MISMATCH0   MISMATCH1</p> <p>Selects the type of pattern error.</p> <p>BIT: Bit</p> <p>MISMATCH0: Mismatch '0'</p> <p>MISMATCH1: Mismatch '1'</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of Pattern error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:PATT:ERR:HIST? 1, BIT</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:HISTory?</code>

---

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:CURRent?**

<b>Description</b>	<p>This query returns the current status of pattern error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Unframed &gt; Results &gt; Unframed Summary (BER)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:CURRent? <wsp> <Lane>, BIT   MISMATCH0   MISMATCH1
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the transmitted error pattern. The range for the lane is from 0 to 19.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT   MISMATCH0   MISMATCH1</p> <p>Selects the type of pattern error.</p> <p>BIT: Bit</p> <p>MISMATCH0: Mismatch '0'</p> <p>MISMATCH1: Mismatch '1'</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of Pattern error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PATT:ERR:CURR? 1, BIT
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:HISTory?

## SCPI Command Reference

### Summary

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE?**

<b>Description</b>	<p>This query returns the Unframed Bert alarm type.</p> <p>At *RST condition, this value is set to PLOSs.</p> <p>Navigation Path: Test (Unframed Bert) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Unframed Bert alarm.</p> <p>PLOSs, PLOSs is selected as the Unframed Bert alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ALAR:TYPE?</p> <p>Returns: PLOS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm</p> <p>SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm?</p>

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm**

<b>Description</b>	<p>This command enables or disables the status of the unframed BERT alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Unframed Bert) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the unframed BERT alarm generation.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ALAR ON</p> <p>SOUR:DATA:TEL:UNFR:PATT:ALAR?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE?

---

## SCPI Command Reference

### Summary

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm?**

<b>Description</b>	<p>This query returns the status of BERT alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Unframed Bert) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Unframed BERT alarm generation.</p> <p>1, returns alarm generation as enabled.</p> <p>0, returns alarm generation as disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ALAR ON</p> <p>SOUR:DATA:TEL:UNFR:PATT:ALAR?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:MANual:TYPE?**

<b>Description</b>	<p>This query returns the type of Unframed Bert error.</p> <p>At *RST condition, this value is set to BIT.</p> <p>Navigation Path: Test (Unframed Bert) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Manual) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the manual type of Unframed Bert error.</p> <p>BIT, bit is selected as manual type of Unframed Bert error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ERR:MAN:TYPE?</p> <p>Returns: BIT</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ALARm:TYPE?

---

## SCPI Command Reference

### Summary

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:MANual:AMOUNT**

<b>Description</b>	<p>This command sets the amount of Unframed Bert error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test (Unframed Bert) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:MANual:AMOUNT &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of Unframed Bert error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ERR:MAN:AMO 25</p> <p>SOUR:DATA:TEL:UNFR:PATT:ERR:MAN:AMO?</p> <p>Returns: 25</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ALARm:TYPE?</p>

---



**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:MANual:AMOunt?**

<b>Description</b>	<p>This query returns the amount of Unframed Bert error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test (Unframed Bert) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:MANual:AMOunt? [<wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Gets the amount of Unframed Bert error.</p> <p>This parameter is optional. If no token is specified, the current amount of Unframed Bert error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of Unframed Bert error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ERR:MAN:AMO 25</p> <p>SOUR:DATA:TEL:UNFR:PATT:ERR:MAN:AMO?</p> <p>Returns: 25</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ALARm:TYPE?

## SCPI Command Reference

### Summary

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:MANual:INJect**

<b>Description</b>	<p>This command injects the Unframed Bert manual error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test (Unframed Bert) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:MANual:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:UNFR:PATT:ERR:MAN:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:TYPE?**

<b>Description</b>	<p>This query gets the automated type of Unframed Bert error.</p> <p>At *RST condition, this value is set to BIT.</p> <p>Navigation Path: Test (Unframed Bert) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the automated type Unframed Bert error.</p> <p>BIT, bit is selected as the automated type Unframed Bert error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT:TYPE?</p> <p>Returns: BIT</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE?

---

## SCPI Command Reference

### Summary

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:AUTomated:RATE**

<b>Description</b>	<p>This command sets the rate of Unframed Bert error.</p> <p>At *RST condition, this value is set to 1.0E-09.</p> <p>Navigation Path: Test (Unframed Bert) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the rate of Unframed Bert error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT:RATE 1.0E-03</p> <p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT:RATE?</p> <p>Returns: 1.0E-03</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ALARm:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the rate of Unframed Bert error.</p> <p>At *RST condition, this value is set to 1.0E-09.</p> <p>Navigation Path: Test (Unframed Bert) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:AUTomated:RATE? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Gets the rate of Unframed Bert error.</p> <p>This parameter is optional. If no token is specified, the current rate of Unframed Bert error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of Unframed Bert error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT:RATE 1.0E-03</p> <p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT:RATE?</p> <p>Returns: 1.0E-03</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ALARm:TYPE?

## SCPI Command Reference

### Summary

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:CONTInuous**

<b>Description</b>	<p>This command sets the continuous mode of the Unframed BERT error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Unframed BERT) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (MaxRate)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status of automated continuous mode of Unframed BERT error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT:CONT ON</p> <p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:AUTomated:CONTInuous?**

<b>Description</b>	<p>This query returns the status of automated continuous mode of the Unframed BERT error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Unframed BERT &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (MaxRate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ERRor:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the automated continuous mode.</p> <p>1, returns the continuous mode status as enabled.</p> <p>0, returns the continuous mode status as disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT:CONT ON</p> <p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATTern:ALARm:TYPE?

---

## SCPI Command Reference

### Summary

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated**

<b>Description</b>	<p>This command sets the automated mode status for the Unframed BERT error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Unframed BERT) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status of automated error injection for the Unframed BERT error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT ON</p> <p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE?



**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated?**

<b>Description</b>	<p>This query returns the automated mode status for the Unframed BERT error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Unframed BERT) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ERRor:AUTomated?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the automated error injection for the Unframed BERT error.</p> <p>1, returns the status of automated error injection as enabled.</p> <p>0, returns the status of automated error injection as disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT ON</p> <p>SOUR:DATA:TEL:UNFR:PATT:ERR:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALARm:TYPE?

---

## SCPI Command Reference

### Summary

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:LANE**

<b>Description</b>	<p>This command sets the particular lane number for the injection purpose for unframed BERT alarms and errors.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Unframed BERT) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Lane</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:LANE &lt;wsp&gt; &lt;Lane&gt;, ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which the injection is to be done for unframed BERT alarms and errors.</p> <p>The range is from 0 to 3 for 4 Lanes and 0 to 19 for 10 lanes</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the per lane status for unframed BERT alarms and errors.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:UNFR:PATT:LANE 5,ON SOUR:DATA:TEL:UNFR:PATT:LANE? 5 Returns: 1</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALANes?</code>

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:LANE?**

---

<b>Description</b>	<p>This query returns the particular lane number for the injection purpose for unframed BERT alarms and errors.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Unframed BERT) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; Lane</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:LANE? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which injection is to be done for unframed BERT alarms and errors. The range is from 0 to 3 for 4 Lanes and 0 to 19 for 10 lanes</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of unframed BERT alarms and errors lane.</p> <p>1, returns the status of particular lane as enabled.</p> <p>0, returns the status of particular lane as disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:LANE 5,ON</p> <p>SOUR:DATA:TEL:UNFR:PATT:LANE? 5</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALANes

---

## SCPI Command Reference

### Summary

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALANes**

<b>Description</b>	<p>This command sets all lanes for the injection purpose for unframed BERT alarms and errors. At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Unframed BERT) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; All Lanes</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALANes &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the all lane status for unframed BERT alarms and errors.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ALAN ON</p> <p>SOUR:DATA:TEL:UNFR:PATT:ALAN?</p> <p>Returns: 1</p>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:LANE?</code>

---

**:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALANes?**

<b>Description</b>	<p>This query returns all lanes for the injection purpose for unframed BERT alarms and errors. At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Unframed BERT) &gt; Results &gt; Summary &gt; Global Injection &gt; Layer (BERT) &gt; All Lanes</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:ALANes?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of unframed BERT alarms and errors for all lanes.</p> <p>1, returns the status of all lanes as enabled.</p> <p>0, returns the status of all lanes as disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:UNFR:PATT:ALAN ON</p> <p>SOUR:DATA:TEL:UNFR:PATT:ALAN?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:UNFRamed:PATtern:LANE

---

## SCPI Command Reference

### Summary

---

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:THReshold:VERDict?**

<b>Description</b>	This query returns the Status of BERT Verdict. At *RST condition, this value is set to BIT. Navigation Path: Test > Results > Summary > BERT
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:THReshold:VERDict?[ <wsp> <Channel>]
<b>Parameter(s)</b>	<b>Channel:</b> The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the channel that will be used by command/query. This parameter can only be used in Multi-Channel OTN. The numeric channel ranges from [1:n] in function of ODU Mapping.
<b>Response Syntax</b>	<STATUS>
<b>Response(s)</b>	<b>STATUS:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the status of Pattern Verdict PASS, Verdict is enabled. FAIL, Verdict is disabled.
<b>Example(s)</b>	FETC:DATA:TEL:PATT:ERR:PATT:THR:VERD? FETC:DATA:TEL:PATT:ERR:PATT:THR:VERD? 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:SEConds?

**:FETCh[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:VERDict?**

<b>Description</b>	<p>This query returns the Status of BERT Verdict.</p> <p>At *RST condition, this value is set to BIT.</p> <p>Navigation Path: Test &gt; Results &gt; Summary &gt; Unframed BERT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:UPRBs:PATtern:THReshold:VERDict? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which the injection is to be done for unframed BERT alarms and errors.</p> <p>The range is from 0 to 3 for 4 Lanes and 0 to 19 for 10 lanes.</p>
<b>Response Syntax</b>	<STATUS>
<b>Response(s)</b>	<p><b>STATUS:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the status of Pattern Unframed BERT Verdict.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	FETC:DATA:TEL:UPRB:PATT:THR:VERD? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:SEConds?

---

## SCPI Command Reference

### Summary

---

**:FETCh[1..n]:DATA:TELeom:SDT:SHORtest?**

<b>Description</b>	<p>This query returns the shortest disruption duration, since test was started/reset. At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;Summary&gt;Service disruption Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:SDT:SHORtest?[ <wsp> <Channel>]
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>This parameter is mandatory in the Multi-Channel OTN application, and cannot be specified in other applications.</p> <p>The query returns the shortest disruption duration for the specified channel.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Shortest>
<b>Response(s)</b>	<p><b>Shortest:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the shortest disruption duration, since test was started/reset.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDT:SHOR?</p> <p>FETC:DATA:TEL:SDT:SHOR? 3</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:SDT:COUNT?

---



**:FETCh[1..n]:DATA:TELEcom:SDT:LONGest?**

<b>Description</b>	<p>This query returns the longest disruption duration, since test was started/reset.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;Summary&gt;Service disruption</p> <p>Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDT:LONGest?[ <wsp> <Channel>]
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>This parameter is optional in the Multi-Channel OTN application, and cannot be specified in other applications.</p> <p>When the channel is specified, the query returns the longest disruption duration for the specified channel.</p> <p>When the channel is specified, the numeric channel ranges from [1:n] in function of ODU Mapping.</p> <p>When the channel is NOT specified, the query returns the longest disruption of ALL monitored channels.</p>
<b>Response Syntax</b>	<Longest>
<b>Response(s)</b>	<p><b>Longest:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the longest disruption duration, since test was started/reset.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDT:LONG?</p> <p>FETC:DATA:TEL:SDT:LONG? 3</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDT:SHORTest?

## SCPI Command Reference

### Summary

---

**:FETCh[1..n]:DATA:TELecom:SDT:LAST?**

<b>Description</b>	<p>This query returns the length of the last disruption time, since test was started/reset.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;Summary&gt;Service disruption</p> <p>Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:SDT:LAST?[ <wsp> <Channel>]
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>This parameter is optional in the Multi-Channel OTN application, and cannot be specified in other applications.</p> <p>When the channel is specified, the query returns the last disruption duration for the specified channel.</p> <p>When the channel is specified, the numeric channel ranges from [1:n] in function of ODU Mapping.</p> <p>When the channel is NOT specified, the query returns the last disruption of ALL monitored channels.</p>
<b>Response Syntax</b>	<Last>
<b>Response(s)</b>	<p><b>Last:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the last disruption duration, since test was started/reset.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDT:LAST?</p> <p>FETC:DATA:TEL:SDT:LAST? 3</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:SDT:LONGest?

---

**:FETCh[1..n]:DATA:TELeom:SDT:AVERAge?**

---

<b>Description</b>	<p>This query returns the average disruption duration, since test was started/reset. At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;Summary&gt;Service disruption Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:SDT:AVERAge?[ <wsp> <Channel>]
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>This parameter is mandatory in the Multi-Channel OTN application, and cannot be specified in other applications.</p> <p>The query returns the average disruption duration for the specified channel.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Average>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the average disruption duration, since test was started/reset.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDT:AVER?</p> <p>FETC:DATA:TEL:SDT:AVER? 3</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:SDT:COUN?

---

## SCPI Command Reference

### Summary

---

**:FETCh[1..n]:DATA:TELecom:SDT:TOTal?**

<b>Description</b>	<p>This query returns the total disruption duration, since test was started/reset. At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;Summary&gt;Service disruption Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:SDT:TOTal?[ <wsp> <Channel>]
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>This parameter is mandatory in the Multi-Channel OTN application, and cannot be specified in other applications.</p> <p>The query returns the total disruption duration for the specified channel.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Total>
<b>Response(s)</b>	<p><b>Total:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the total disruption duration, since test was started/reset.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDT:TOT? FETC:DATA:TEL:SDT:TOT? 3</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:SDT:LAST?

---

**:FETCh[1..n]:DATA:TELeom:SDT:COUNT?**

<b>Description</b>	<p>This query returns the number of service disruptions, since test was started/reset.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;Summary&gt;Service disruption</p> <p>Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:SDT:COUNT?[ <wsp> <Channel>]
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>This parameter is mandatory in the Multi-Channel OTN application, and cannot be specified in other applications.</p> <p>The query returns the number of service disruption for the specified channel.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of service disruptions, since test was started/reset.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDT:COUN?</p> <p>FETC:DATA:TEL:SDT:COUN? 3</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:SDT:TOTAl?

## SCPI Command Reference

### Summary

---

**:FETCh[1..n]:DATA:TELEcom:SDT:DEFect?**

<b>Description</b>	This query returns the layer on which service disruption time test is performed for OTN. At *RST condition, this value is device dependent. Navigation Path: Test > OTN BERT >Results>Summary>Service disruption>Defect
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDT:DEFect?
<b>Response Syntax</b>	<Defect>
<b>Response(s)</b>	<b>Defect:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. This query returns the layer on which service disruption time test is performed for OTN.
<b>Example(s)</b>	FETC:DATA:TEL:SDT:DEF?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTL:SDT:DEFect?

---

---

**:FETCh[1..n]:DATA:TELecom:SDT:VERDict?**

<b>Description</b>	<p>This query returns if a SDT disruption greater than than the Pass/Fail verdict threshold was detected.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;Summary&gt;Service disruption&gt;Defect</p> <p>Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:SDT:VERDict?[ <wsp> <Channel>]
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>This parameter is mandatory in the Multi-Channel OTN application, and cannot be specified in other applications.</p> <p>The query returns if a SDT disruption greater than than the Pass/Fail verdict threshold was detected for the specified channel.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<STATUS>
<b>Response(s)</b>	<p><b>STATUS:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the status of Pattern SDT Verdict.</p> <p>NONE, verdict is not present.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDT:VERD?</p> <p>FETC:DATA:TEL:SDT:VERD? 3</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTL:SDT:DEFect?</p> <p>FETCh[1..n]:DATA:TELecom:SDT:CHAT?</p>

---

## SCPI Command Reference

### Summary

---

---

#### :FETCh[1..n]:DATA:TELecom:OTL:SUMMArY:SDT:LONGest?

---

<b>Description</b>	<p>This query returns the longest disruption time for summary in OTL multilane configuration. At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results&gt;Summary&gt;Service disruption &gt;Disruption Time</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTL:SUMMArY:SDT:LONGest?
<b>Response Syntax</b>	<Longest>
<b>Response(s)</b>	<p><b>Longest:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the longest disruption time.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SUMMArY:SDT:LONG?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:SDT:COUNT?

---



**:FETCh[1..n]:DATA:TELEcom:OTL:SUMMAry:SDT:SHORtest?**

<b>Description</b>	<p>This query returns the shortest disruption time for summary in OTL multilane configuration. At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results&gt;Summary&gt;Service disruption &gt;Disruption Time</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTL:SUMMAry:SDT:SHORtest?
<b>Response Syntax</b>	<Shortest>
<b>Response(s)</b>	<p><b>Shortest:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the shortest disruption time.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SUMMAry:SDT:SHOR?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDT:LONG?

---

## SCPI Command Reference

### Summary

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTL:SUMMery:SDT:LAST?

---

<b>Description</b>	<p>This query returns the length of the last disruption time for summary in OTL multilane configuration.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results&gt; Summary&gt;Service disruption &gt;Disruption Time</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTL:SUMMery:SDT:LAST?
<b>Response Syntax</b>	<Last>
<b>Response(s)</b>	<p><b>Last:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the last disruption time.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SUMMery:SDT:LAST?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDT:LAST?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTL:SUMMary:SDT:AVERage?**

---

<b>Description</b>	<p>This query returns the average disruption time for summary in OTL multilane configuration. At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT&gt; Results&gt;Summary&gt;Service disruption &gt;Disruption Time</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTL:SUMMary:SDT:AVERage?
<b>Response Syntax</b>	<Average>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the average disruption time.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SUMMary:SDT:AVER?
<b>See Also</b>	FETCh[1..n]:DATA:TEL:OTL:SDT:DEF?

---

## SCPI Command Reference

### Summary

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTL:SUMMery:SDT:TOTal?

---

<b>Description</b>	<p>This query returns the total disruption time for summary in OTL multilane configuration.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT&gt;Results&gt;Summary&gt;Service disruption &gt;Disruption Time</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTL:SUMMery:SDT:TOTal?
<b>Response Syntax</b>	<Total>
<b>Response(s)</b>	<p><b>Total:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the total disruption time.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SUMMery:SDT:TOT?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDT:COUNT?

---

**:FETCh[1..n]:DATA:TELEcom:OTL:SUMMary:SDT:COUNt?**

<b>Description</b>	<p>This query returns the number of service disruption counts that happened since the beginning of the Service Disruption Time (SDT) test for summary in OTL multilane configuration.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT&gt; Results&gt; Summary&gt; Service disruption&gt;Disruption Time</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTL:SUMMary:SDT:COUNt?
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of service disruption counts that happened since the beginning of the SDT test.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SUMMary:SDT:COUN?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDT:TOTal?

---

## Summary (Traffic Gen & Mon) - (Stream)

---

:FETCh[1..n]:DATA:TELEcom:ETHernet:COUNT:FRAMES:RX?

---

<b>Description</b>	<p>This query returns the number of frames received matching the selected stream.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Summary &gt; Stream (Pop up) &gt;RX Frame Count</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:COUNT:FRAMES:RX? <wsp> <Tgen>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<Framerx>
<b>Response(s)</b>	<p><b>Framerx:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of flow control pause frames received.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:COUN:FRAM:RX? 2
<b>See Also</b>	SOURce[1..n]:DATA:TEL:ETH:STR:PAYL

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:COUNt:FRAMes:TX?**

---

<b>Description</b>	<p>This query returns the frames transmitted for selected traffic stream.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Summary &gt; Stream (Pop up) &gt;TX Frame Count</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:COUNt:FRAMes:TX? &lt;wsp&gt; &lt;Tgen&gt;</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;Frame TX&gt;</p>
<b>Response(s)</b>	<p><b>Frame TX:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of flow control pause frames Transmitted.</p>
<b>Example(s)</b>	<p>FETCh:DATA:TEL:ETHernet:COUNt:FRAM:TX? 2</p>
<b>See Also</b>	<p>FETC[1..n]:DATA:TEL:ETH:COUN:FRAM:RX?</p>

---

## Summary (EtherSAM)

---

:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:STATus?

---

<b>Description</b>	<p>This query returns Service CONFig and Service performance test status.</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Summary</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:STATus? &lt;wsp&gt;LTOR   RTOL, SCONTest   SPERTest</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR, sets the direction from local to remote.</p> <p>RTOL, sets the direction from remote to local.</p> <p><b>Test:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SCONTest   SPERTest</p> <p>Selects the type of EtherSAM sub-tests.</p> <p>SCONTest: Service CONFig Test</p> <p>SPERTest: Service performance Test</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Status</p>
<b>Example(s)</b>	<p>FETC:DATA:TELEcom:ETHernet:ESAM:TESTs:STATus? LTOR,SCONTest</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCONTest:BURSt:FLOs:VERDict?</p>

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:VERDict?**

<b>Description</b>	This query returns Service CONFig and Service performance test Verdict. Navigation Path:Test Setup >EtherSAM>Results>Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:VERDict? <wsp>LTOR   RTOL, SCONTest   SPERTest
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR, sets the direction from local to remote.</p> <p>RTOL, sets the direction from remote to local.</p> <p><b>Test:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SCONTest   SPERTest</p> <p>Selects the type of EtherSAM sub-tests.</p> <p>SCONTest: Service CONFig Test</p> <p>SPERTest: Service performance Test</p>
<b>Response Syntax</b>	<Verdict>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Verdict</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	FETC:DATA:TELEcom:ETHernet:ESAM:TESTs:VERDict? LTOR,SCONTest
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:FRASize:QUANtity SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:SERVices:FRASize:QUANtity?

## SCPI Command Reference

### Summary (EtherSAM)

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:VERDict?

---

<b>Description</b>	<p>This query returns Service CONFig test Verdict.</p> <p>Navigation Path:Test Setup &gt;EtherSAM&gt;Results&gt;Service CONFig Test</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:VERDict? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Verdict&gt;</p>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Verdict</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:VERDict? 1, LTOR</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:ADD</p>

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SPRTest:VERDict?**

<b>Description</b>	<p>This query returns Service performance test Verdict.</p> <p>Navigation Path:Navigation Path:Test Setup &gt;EtherSAM&gt;Results&gt;Service Performance Test</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SPRTest:VERDict? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Verdict&gt;</p>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Verdict</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SPRTest:VERDict? 1, LTOR</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DELeTe</p>

---

## SCPI Command Reference

### Summary (EtherSAM)

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOSs?

---

<b>Description</b>	<p>This query returns the percentage of frames that are lost.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Configuration Test&gt;Service Configuration Test&gt;Committed&gt;Frame Loss(%)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOSs? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Frame Loss&gt;</p>
<b>Response(s)</b>	<p><b>Frame Loss:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns frame loss percentage</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOSs? 1, LTOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:FLOSs?</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter?**

<b>Description</b>	<p>This query returns maximum measured delay variation.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Configuration Test&gt;Service Configuration Test&gt;Committed&gt;Max Jitter(ms)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter? <wsp><Service>, LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Max Jitter>
<b>Response(s)</b>	<p><b>Max Jitter:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns Max Jitter</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter? 1, LTOR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:MAXJitter?

## SCPI Command Reference

### Summary (EtherSAM)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency?**

<b>Description</b>	<p>This query returns maximum measured round trip latency (delay).</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Configuration Test&gt;Service Configuration Test&gt;Committed&gt;Max Latency(ms)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Max Latency&gt;</p>
<b>Response(s)</b>	<p><b>Max Latency:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns Max Latency</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency? 1, LTOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:MLATency?</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXRate?**

<b>Description</b>	<p>This query returns measured maximum utilization throughput.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Configuration Test&gt;Service Configuration Test&gt;Excess&gt;Max RX Rate(%)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXRate? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Max Rx Rate&gt;</p>
<b>Response(s)</b>	<p><b>Max Rx Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns Max Rx Rate</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXR? 1, LTOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:ARXRate?</p>

## SCPI Command Reference

### Summary (EtherSAM)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOSs:VERDict?**

<b>Description</b>	<p>This query returns the Frame Loss verdict status.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Configuration Test&gt;Service Configuration Test&gt;Committed&gt;Frame Loss(%)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOSs:VERDict? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Verdict&gt;</p>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Frame Loss verdict status.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOSs:VERDict? 1, LTOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:FLOSs:VERDict?</p>



**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter:VERDict?**

<b>Description</b>	<p>This query returns Max Jitter verdict status.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Configuration Test&gt;Service Configuration Test&gt;Committed&gt;Max Jitter(ms)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter:VERDict ? <wsp> <Service>, LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Verdict>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Max Jitter verdict status.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter:VERDict? 1, LTOR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:MAXJitter:VERDict?

## SCPI Command Reference

### Summary (EtherSAM)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency:VERDict?**

<b>Description</b>	<p>This query returns Max Latency verdict status.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Configuration Test&gt;Service Configuration Test&gt;Committed&gt;Max Latency(ms)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency:VERDict ? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local.</p>
<b>Response Syntax</b>	<p>&lt;Verdict&gt;</p>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Max Latency verdict status.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency:VERDict? 1, LTOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:MLATency:VERDict?</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXRate:VERDict?**

<b>Description</b>	<p>This query returns Max Rx Rate verdict.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Configuration Test&gt;Service Configuration Test&gt;Committed&gt;Max RX Rate(%)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXRate:VERDict? <wsp> <Service>, LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Verdict>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Max Rx Rate verdict status.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXR:VERDict? 1, LTOR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:ARXRRate:VERDict?

## SCPI Command Reference

### Summary (EtherSAM)

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:FLOSs?

---

<b>Description</b>	<p>This query returns the percentage of frames that are lost.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Performance Test&gt;Service Performance Test&gt;Excess&gt;Frame Loss(%)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:FLOSs? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Frame Loss&gt;</p>
<b>Response(s)</b>	<p><b>Frame Loss:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns frame loss percentage</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:FLOSs? 1, LTOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:FLOSs?</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:MAXJitter?**

<b>Description</b>	<p>This query returns the maximum measured delay variation.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Performance Test&gt;Service Performance Test&gt;Committed&gt;Max Jitter(ms)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:MAXJitter? <wsp> <Service>, LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Max Jitter>
<b>Response(s)</b>	<p><b>Max Jitter:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns Max Jitter</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:MAXJitter? 1, LTOR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter?

## SCPI Command Reference

### Summary (EtherSAM)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMARY:SPRTTest:MLATency?**

<b>Description</b>	<p>This query returns the maximum measured round trip latency (delay).</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Performance Test&gt;Service Performance Test&gt;Committed&gt;Max Latency(ms)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMARY:SPRTTest:MLATency? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Max Latency&gt;</p>
<b>Response(s)</b>	<p><b>Max Latency:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns Max Latency</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMARY:SPRT:MLATency? 1, LTOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMARY:SCOTest:MLATency?</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:ARXRate?**

<b>Description</b>	<p>This query returns the average utilization throughput.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Performance Test&gt;Service Performance Test&gt;Committed&gt;Avg RX Rate(%)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:ARXRate? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Avg Rx Rate&gt;</p>
<b>Response(s)</b>	<p><b>Avg Rx Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns Average Rx Rate</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:ARXRate? 1, LTOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXR?</p>

## SCPI Command Reference

### Summary (EtherSAM)

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMery:SPRTest:FLOSs:VERDict?

---

<b>Description</b>	<p>This query returns the percentage of Frame Loss verdict.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Performance Test&gt;Service Performance Test&gt;Committed&gt;Frame Loss(%)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMery:SPRTest:FLOSs:VERDict? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Verdict&gt;</p>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Frame Loss verdict status.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMery:SPRT:FLOSs:VERDict? 1, LTOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMery:SCOTest:FLOSs:VERDict?</p>

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:MAXJitter:VERDict?**

<b>Description</b>	<p>This query returns the Max Jitter verdict.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Performance Test&gt;Service Performance Test&gt;Committed&gt;Max Jitter(ms)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:MAXJitter:VERDict? <wsp> <Service>, LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Verdict>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Max Jitter verdict status.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:MAXJitter:VERDict? 1, LTOR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MAXJitter:VERDict?

## SCPI Command Reference

### Summary (EtherSAM)

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:MLATency:VERDict?

---

<b>Description</b>	<p>This query returns the Max Latency verdict.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Performance Test&gt;Service Performance Test&gt;Committed&gt;Max Latency(ms)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:MLATency:VERDict ? <wsp> <Service>, LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Verdict>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Max Latency verdict status.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:MLATency:VERDict? 1, LTOR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MLATency:VERDict ?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:ARXRate:VERDict?**

<b>Description</b>	<p>This query returns Avg. Rx Verdict.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Performance Test&gt;Service Performance Test&gt;Committed&gt;Avg RX Rate(%)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRTTest:ARXRate:VERDict? <wsp> <Service>, LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Verdict>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Avg Rx Rate verdict status.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SPRT:ARXRate:VERDict? 1, LTOR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SCOTest:MRXR:VERDict?

## SCPI Command Reference

### Summary (EtherSAM)

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:VLAN:PREServ?

---

<b>Description</b>	<p>This query returns the Virtual Local Area Network (VLAN) preservation status.</p> <p>At *RST condition, this value is set to UNDEFINED.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Summary&gt;Service Performance Test&gt;Vlan Preservation</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:VLAN:PREServ?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns VLAN preservation status.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:VLAN:PREServ?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM: CONFig:BURSt:PARAMeters:BIRFrame?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SERVices:VLAN:PREserv?**

<b>Description</b>	<p>This query returns the Virtual Local Area Network (VLAN) preservation status for service. At *RST condition, this value is set to UNDEFINED.</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Summary &gt; Service Performance Test &gt; Service</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SUMMary:SERVices:VLAN:PREserv? <wsp> <Service>, LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR and RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns VLAN preservation status.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SUMMary:SERVices:VLAN:PREserv? 1,LTOR
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:CONFig:BURSt:PARAMeters:BIRFrame?

# Summary (RFC 2544)

---

:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TSTate?

---

<b>Description</b>	<p>This query returns the throughput test state.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TSTate?
<b>Response Syntax</b>	<State>
<b>Response(s)</b>	<p><b>State:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the throughput test state.</p> <p>--, indicates that the test has not run yet.</p> <p>INPROGRESS, indicates that the test is running.</p> <p>COMPLETED, indicates that the test is completed.</p> <p>ABORTED, indicates that the test has been interrupted (stopped).</p> <p>INACTIVE, indicates that the test is inactive.</p> <p>FAILED, indicates that the test has been failed.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:THR:TST?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TSTate?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THROughput:SMESsage?**

<b>Description</b>	<p>This query returns the throughput test status messages.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THROughput:SMESsage?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the throughput test status messages.</p> <p>NONE, None is retrieved.</p> <p>INITIALIZING, Initializing is retrieved.</p> <p>WAITING, Waiting is retrieved.</p> <p>NMEASURABLE, Not Measurable is retrieved.</p> <p>SLFRAMES, Sending Learning Frames is retrieved.</p> <p>STFRAMES, Sending Test Frames is retrieved.</p> <p>ABUSER, Aborted by User is retrieved.</p> <p>LDOWN, Link Down is retrieved.</p> <p>ALRCONNECTION, Aborted - Loss of Remote Connection is retrieved.</p> <p>MANRESOLVED, MAC Address Not Resolved is retrieved.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:THR:SMES?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:SMESsage?

---

## SCPI Command Reference

Summary (RFC 2544)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:FCOunt:TX?**

<b>Description</b>	<p>This query returns the number of transmitted frames for the indicated direction.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:FCOunt:TX? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p>
<b>Response Syntax</b>	<p>&lt;Fcount&gt;</p>
<b>Response(s)</b>	<p><b>Fcount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the number of transmitted frames for the indicated direction.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:THR:FCO:TX? TX2RX</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:FCOunt:RX?</p>



**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THRoughput:FCOunt:RX?**

<b>Description</b>	<p>This query returns the number of received frames for the indicated direction.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THRoughput:FCOunt:RX? <wsp>TX2RX   LTORemote   RTOLocal
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p>
<b>Response Syntax</b>	<Fcount>
<b>Response(s)</b>	<p><b>Fcount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the number of received frames for the indicated direction.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:THR:FCO:RX? TX2RX
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THRoughput:FCOunt:TX?

---

## SCPI Command Reference

Summary (RFC 2544)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TRESults[1..n]?**

### Description

This query returns the number of frames for the corresponding direction depending on the selected Layer and Displayed Results.

At \*RST condition, this value is set to device-dependent.

Navigation Path: Test > Results > Summary

### Syntax

:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TRESults[1..n]?  
<wsp>TX2RX | LTORemote | RTOLocal, CURRent | MINimum | MAXimum | AVERage,  
ETHERNET | IP | ALL

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THROUGHput:TRESults[1..n]?**

<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <p><b>Results:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CURRent   MINimum   MAXimum   AVERAge</p> <p>Selects the result mode.</p> <p>CURRent: Current</p> <p>MINimum: Minimum</p> <p>MAXimum: Maximum</p> <p>AVERAge: Average</p> <p><b>Layers:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: ETHERNET   IP   ALL</p> <p>Selects the layers used to calculate the Throughput test.</p> <p>ETHERNET: Ethernet</p> <p>IP: IP</p> <p>ALL: All</p>
<b>Response Syntax</b>	<Results>

## SCPI Command Reference

*Summary (RFC 2544)*

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THROUGHput:TRESults[1..n]?**

<b>Response(s)</b>	<b>Results:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Throughput results.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:THR:TRES1? TX2RX,CURR,ALL
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:FLOs:FRESults[1..n]?

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THRoughput:CTRial?**

---

<b>Description</b>	This query returns the current trial number for the test. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THRoughput:CTRial?
<b>Response Syntax</b>	<Number>
<b>Response(s)</b>	<b>Number:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the current trial number.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:THR:CTR?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:BCKTobck:CTRial?

---

## SCPI Command Reference

Summary (RFC 2544)

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:BCKTobck:TState?**

---

<b>Description</b>	<p>This query returns the back-to-back test state.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:BCKTobck:TState?
<b>Response Syntax</b>	<State>
<b>Response(s)</b>	<p><b>State:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the test state.</p> <p>--, indicates that the test has not run yet.</p> <p>INPROGRESS, indicates that the test is running.</p> <p>COMPLETED, indicates that the test is completed.</p> <p>ABORTED, indicates that the test has been interrupted (stopped).</p> <p>INACTIVE, indicates that the test is inactive.</p> <p>FAILED, indicates that the test has been failed.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:BCKT:TST?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THRoughput:TState?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:SMESsage?**

<b>Description</b>	<p>This query returns the back-to-back test status messages.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:SMESsage?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the test status.</p> <p>NONE, None is retrieved.</p> <p>INITIALIZING, Initializing is retrieved.</p> <p>WAITING, Waiting is retrieved.</p> <p>NMEASURABLE, Not Measurable is retrieved.</p> <p>SLFRAMES, Sending Learning Frames is retrieved.</p> <p>STFRAMES, Sending Test Frames is retrieved.</p> <p>ABUSER, Aborted by User is retrieved.</p> <p>LDOWN, Link Down is retrieved.</p> <p>ALRCONNECTION, Aborted - Loss of Remote Connection is retrieved.</p> <p>MANRESOLVED, MAC Address Not Resolved is retrieved.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:BCKT:SMES?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:SMESsage?

---

## SCPI Command Reference

Summary (RFC 2544)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:FCOunt:TX?**

<b>Description</b>	<p>This query returns the number of transmitted frames for the indicated direction.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:FCOunt:TX? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p>
<b>Response Syntax</b>	<p>&lt;Fcount&gt;</p>
<b>Response(s)</b>	<p><b>Fcount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the number of transmitted frames for the indicated direction.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:BCKT:FCO:TX? TX2RX</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:FCOunt:RX?</p>



**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:FCOunt:RX?**

<b>Description</b>	<p>This query returns the number of received frames for the indicated direction.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:FCOunt:RX? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal</code>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p>
<b>Response Syntax</b>	<code>&lt;Fcount&gt;</code>
<b>Response(s)</b>	<p><b>Fcount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the number of received frames for the indicated direction.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:ETH:RFC:BCKT:FCO:RX? TX2RX</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:FCOunt:TX?</code>

---

## SCPI Command Reference

Summary (RFC 2544)

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:BCKTobck:CTRial?**

---

<b>Description</b>	This query returns the current trial number for the test. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:BCKTobck:CTRial?
<b>Response Syntax</b>	<Number>
<b>Response(s)</b>	<b>Number:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the current trial number for the test.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:BCKT:CTR?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THRoughput:CTRial?

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:BBResults[1..n]?**

<b>Description</b>	<p>This query returns the number of frames for the corresponding direction depending on the selected Layer and Displayed Results.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:BBResults[1..n]? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal, CURRent   MINimum   MAXimum   AVERage, ETHERNET   IP   ALL</p>

---

## SCPI Command Reference

Summary (RFC 2544)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKToBck:BBResults[1..n]?**

<b>Parameter(s)</b>	<b>Direction:</b> The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal Selects the direction. TX2RX: TX-to-RX (TX2RX) for single port topology. LTORemote: Local-to-Remote direction. RTOLocal: Remote-to-Local direction. <b>Results:</b> The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are: CURRent   MINimum   MAXimum   AVERage Selects the result mode. CURRent: Current MINimum: Minimum MAXimum: Maximum AVERage: Average <b>Layers:</b> The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are: ETHERNET   IP   ALL Selects the layers used to calculate the back-to-back test. ETHERNET: Ethernet IP: IP ALL: All
<b>Response Syntax</b>	<Results>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:BBResults[1..n]?**

**Response(s)**      **Results:**

The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element.

Returns the back-to-back results.

TX2RX, returns TX-to-RX (TX2RX) for single port topology.

**Example(s)**      FETC:DATA:TEL:ETH:RFC:BCKT:BBR1? TX2RX,CURR,ALL

**See Also**      FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:CTRial?

---

## SCPI Command Reference

Summary (RFC 2544)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOsS:TSTate?**

<b>Description</b>	<p>This query returns the Frame Loss test state.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOsS:TSTate?
<b>Response Syntax</b>	<State>
<b>Response(s)</b>	<p><b>State:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Frame Loss test state.</p> <p>--, indicates that the test has not run yet.</p> <p>INPROGRESS, indicates that the test is running.</p> <p>COMPLETED, indicates that the test is completed.</p> <p>ABORTED, indicates that the test has been interrupted (stopped).</p> <p>INACTIVE, indicates that the test is inactive.</p> <p>FAILED, indicates that the test has been failed.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:FLOS:TST?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TSTate?

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:SMESsage?**

---

<b>Description</b>	<p>This query returns the Frame Loss test status messages.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:SMESsage?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Frame Loss test status.</p> <p>NONE, None is retrieved.</p> <p>INITIALIZING, Initializing is retrieved.</p> <p>WAITING, Waiting is retrieved.</p> <p>NMEASURABLE, Not Measurable is retrieved.</p> <p>SLFRAMES, Sending Learning Frames is retrieved.</p> <p>STFRAMES, Sending Test Frames is retrieved.</p> <p>ABUSER, Aborted by User is retrieved.</p> <p>LDOWN, Link Down is retrieved.</p> <p>ALRCONNECTION, Aborted - Loss of Remote Connection is retrieved.</p> <p>MANRESOLVED, MAC Address Not Resolved is retrieved.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:FLOS:SMES?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:SMESsage?

---

## SCPI Command Reference

Summary (RFC 2544)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:FCOunt:TX?**

<b>Description</b>	<p>This query returns the number of transmitted frames for the indicated direction.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:FCOunt:TX? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p>
<b>Response Syntax</b>	<p>&lt;Fcount&gt;</p>
<b>Response(s)</b>	<p><b>Fcount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the number of transmitted frames for the indicated direction.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:FLOS:FCO:TX? TX2RX</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOs:FCOunt:RX?</p>

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:FCOunt:RX?**

<b>Description</b>	<p>This query returns the number of received frames for the indicated direction.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:FCOunt:RX? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p>
<b>Response Syntax</b>	<p>&lt;Fcount&gt;</p>
<b>Response(s)</b>	<p><b>Fcount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the number of received frames for the indicated direction.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:FLOS:FCO:RX? TX2RX</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:FCOunt:TX?</p>

## SCPI Command Reference

Summary (RFC 2544)

---

---

**:FETCh[1..n]:DATA:TELeom:ETHernet:RFC:FLOs:CTRial?**

---

<b>Description</b>	This query returns the current trial number for the test. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:ETHernet:RFC:FLOs:CTRial?
<b>Response Syntax</b>	<Number>
<b>Response(s)</b>	<b>Number:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the current trial number for the test.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:FLOS:CTR?
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:ETHernet:RFC:BCKTobck:CTRial?

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:CSTep?**

---

<b>Description</b>	This query returns the current percentage of the testing rate. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:CSTep?
<b>Response Syntax</b>	<Step>
<b>Response(s)</b>	<b>Step:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the current percentage of the testing rate.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:FLOS:CST?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:CTRial?

---

## SCPI Command Reference

Summary (RFC 2544)

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:FLOsS:FRESult[1..n]?**

<b>Description</b>	<p>This query returns the number of frames for the corresponding direction depending on the selected Displayed Step and Displayed Results.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:FLOsS:FRESult[1..n]? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal, CURRent   MINimum   MAXimum   AVERage, &lt;Step&gt;</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <p><b>Results:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CURRent   MINimum   MAXimum   AVERage</p> <p>Selects the result mode.</p> <p>CURRent: Current</p> <p>MINimum: Minimum</p> <p>MAXimum: Maximum</p> <p>AVERage: Average</p> <p><b>Step:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Sets the current percentage of the testing rate.</p>
<b>Response Syntax</b>	<p>&lt;Results&gt;</p>

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:FLOs:FRESult[1..n]?**

---

<b>Response(s)</b>	<b>Results:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Frame Loss results.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:FLOS:FRES1? TX2RX,CURR,90.000
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THRoughput:TRESults[1..n]?

---

## SCPI Command Reference

Summary (RFC 2544)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TSTate?**

<b>Description</b>	<p>This query returns the latency test state.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TSTate?
<b>Response Syntax</b>	<State>
<b>Response(s)</b>	<p><b>State:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the latency test state.</p> <p>--, indicates that the test has not run yet.</p> <p>INPROGRESS, indicates that the test is running.</p> <p>COMPLETED, indicates that the test is completed.</p> <p>ABORTED, indicates that the test has been interrupted (stopped).</p> <p>INACTIVE, indicates that the test is inactive.</p> <p>FAILED, indicates that the test has been failed.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:LAT:TST?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOsS:TSTate?

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:LATency:SMESsage?**

<b>Description</b>	<p>This query returns the latency test status messages.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:LATency:SMESsage?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the latency test status.</p> <p>NONE, None is retrieved.</p> <p>INITIALIZING, Initializing is retrieved.</p> <p>WAITING, Waiting is retrieved.</p> <p>NMEASURABLE, Not Measurable is retrieved.</p> <p>SLFRAMES, Sending Learning Frames is retrieved.</p> <p>STFRAMES, Sending Test Frames is retrieved.</p> <p>ABUSER, Aborted by User is retrieved.</p> <p>LDOWN, Link Down is retrieved.</p> <p>ALRCONNECTION, Aborted - Loss of Remote Connection is retrieved.</p> <p>MANRESOLVED, MAC Address Not Resolved is retrieved.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:LAT:SMES?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:FLOSs:SMESsage?

---

## SCPI Command Reference

Summary (RFC 2544)

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:CTRial?**

---

<b>Description</b>	This query returns the current trial number for the test. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:CTRial?
<b>Response Syntax</b>	<Number>
<b>Response(s)</b>	<b>Number:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the current trial number for the test.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:LAT:CTR?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:FLOSs:CTRial?

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:LRESults[1..n]?**

<b>Description</b>	<p>This query returns the number of frames for the corresponding direction depending on the selected Mode and Displayed Results.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:LRESults[1..n]? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal, CURRent   MINimum   MAXimum   AVERage, SFORward   CTHROUGH</p>

---

## SCPI Command Reference

Summary (RFC 2544)

---

:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:LRESults[1..n]?

<b>Parameter(s)</b>	<b>Direction:</b> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p> <b>Results:</b> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CURRent   MINimum   MAXimum   AVERAge</p> <p>Selects the result mode.</p> <p>CURRent: Current</p> <p>MINimum: Minimum</p> <p>MAXimum: Maximum</p> <p>AVERAge: Average</p> <b>Mode:</b> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SFORward   CTHROUGH</p> <p>Selects the propagation time mode.</p> <p>SFORward: Store and Forward (SFORward), which allows the calculation of the propagation time of a frame.</p> <p>CTHROUGH: Cut Through (CTHROUGH), which allows the calculation of the propagation time of a bit.</p>
---------------------	---

**Response Syntax** <Results>

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:LATency:LRESults[1..n]?**

---

<b>Response(s)</b>	<b>Results:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the Latency results.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:LAT:LRES1? TX2RX,CURR,CTHR
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:THroughput:TRESults[1..n]?

---

## SCPI Command Reference

Summary (RFC 2544)

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:LATency:FCOunt:TX?**

---

<b>Description</b>	<p>This query returns the number of transmitted frames for the indicated direction.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:LATency:FCOunt:TX? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p>
<b>Response Syntax</b>	<p>&lt;Fcount&gt;</p>
<b>Response(s)</b>	<p><b>Fcount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the number of transmitted frames for the indicated direction.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:LAT:FCO:TX? TX2RX</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:ETHernet:RFC:LATency:FCOunt:RX?</p>

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:LATency:FCOunt:RX?**

<b>Description</b>	<p>This query returns the number of received frames for the indicated direction.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:LATency:FCOunt:RX? &lt;wsp&gt;TX2RX   LTORemote   RTOLocal</code>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TX2RX   LTORemote   RTOLocal</p> <p>Selects the direction.</p> <p>TX2RX: TX-to-RX (TX2RX) for single port topology.</p> <p>LTORemote: Local-to-Remote direction.</p> <p>RTOLocal: Remote-to-Local direction.</p>
<b>Response Syntax</b>	<code>&lt;Fcount&gt;</code>
<b>Response(s)</b>	<p><b>Fcount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the number of received frames for the indicated direction.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:ETH:RFC:LAT:FCO:RX? TX2RX</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELecom:ETHernet:RFC:LATency:FCOunt:TX?</code>

---

## SCPI Command Reference

Summary (RFC 2544)

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TETime?**

---

<b>Description</b>	This query returns the elapsed time for the Latency subtest. At *RST condition, this value is set to device-dependent. Navigation Path: Result > Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:TETime?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<b>Time:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the elapsed time.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:LAT:TET?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TETime?

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TETime?**

---

<b>Description</b>	This query returns the elapsed time for the back-to-back subtest. At *RST condition, this value is set to device-dependent. Navigation Path: Result > Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TETime?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<b>Time:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the elapsed time.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:BCKT:TET?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TETime?

---

## SCPI Command Reference

Summary (RFC 2544)

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TETime?**

---

<b>Description</b>	This query returns the elapsed time for the throughput subtest. At *RST condition, this value is set to device-dependent. Navigation Path: Result > Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:THRoughput:TETime?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<b>Time:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the elapsed time.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:THR:TET?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:BCKTobck:TETime?

---



---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:FLOs:TETime?**

---

<b>Description</b>	This query returns the elapsed time for the Frame Loss subtest. At *RST condition, this value is set to device-dependent. Navigation Path: Result > Summary
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:FLOs:TETime?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<b>Time:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the elapsed time.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:FLOS:TET?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:BCKTobck:TETime?

---

## SCPI Command Reference

Summary (RFC 2544)

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:SUMMary:THReshold:VERDict?**

<b>Description</b>	<p>This query returns verdict status depends upon test and frame size.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:SUMMary:THReshold:VERDict? &lt;wsp&gt;THR   BTB   FLOS   LAT, F64   F128   F256   F512   F1024   F1280   F1518</p>
<b>Parameter(s)</b>	<p><b>VALue:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: THR   BTB   FLOS   LAT</p> <p>Selects the Subtests</p> <p>THR: Throughput test.</p> <p>BTB: Back-to-back Test.</p> <p>FLOS: Frame Loss Test.</p> <p>LAT: Latency test.</p> <p><b>Fsize:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: F64   F128   F256   F512   F1024   F1280   F1518</p> <p>Selects the Frame Size</p> <p>F64: 64</p> <p>F128: 128</p> <p>F256: 256</p> <p>F512: 512</p> <p>F1024: 1024</p> <p>F1280: 1280</p> <p>F1518: 1518</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:SUMMary:THReshold:VERDict?**

<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the verdict status. None, returns test is pending. Pass, returns test is pass. Fail, returns test is fail.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:SUMM:THR:VERD? THR,F1518
<b>See Also</b>	FETCh[1..n]:DATA:TEL:SON:LINE:PM:STAT?

---

## SCPI Command Reference

Summary (RFC 2544)

---

### :FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:THReshold:VERDict?

<b>Description</b>	<p>This query returns verdict status.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Summary</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:GLOBal:THReshold:VERDict? <wsp>THR   BTB   FLOS   LAT
<b>Parameter(s)</b>	<p><b>VALue:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: THR   BTB   FLOS   LAT</p> <p>Selects the Subtests.</p> <p>THR: Throughput test.</p> <p>BTB: Back-to-back Test.</p> <p>FLOS: Frame Loss Test.</p> <p>LAT: Latency test.</p>
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the verdict status.</p> <p>None, returns test is pending.</p> <p>Pass, returns test is pass.</p> <p>Fail, returns test is fail.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:GLOB:THR:VERD? THR
<b>See Also</b>	FETCh[1..n]:DATA:TEL:SON:LINE:PM:STAT?

## Summary (Traffic Gen & Mon)

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:STATus:TIME?**

---

<b>Description</b>	<p>This query returns the time required for a bit to travel from the transmitter back to its receiver. after crossing a far-end loopback.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary &gt; QoS Metrics</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:STATus:TIME? &lt;wsp&gt;CURRENT   LATMIN   LATMAX   AVERAge</p>
<b>Parameter(s)</b>	<p><b>Time:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CURRENT   LATMIN   LATMAX   AVERAge</p> <p>Sets the time required for a bit to travel.</p> <p>CURRENT, displays the result of the last latency measurement.</p> <p>LATMIN, displays the minimum latency recorded.</p> <p>LATMAX, displays the maximum latency recorded.</p> <p>AVERAge, displays the average latency value.</p>
<b>Response Syntax</b>	<p>&lt;Time&gt;</p>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the time required for a bit to travel.</p> <p>CURRENT, displays the result of the last latency measurement.</p> <p>LATMIN, displays the minimum latency recorded.</p> <p>LATMAX, displays the maximum latency recorded.</p> <p>AVERAGE, displays the average latency value.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:STR:LAT:STAT:TIME? CURR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:SECOnds?</p>

---

## SCPI Command Reference

Summary (Traffic Gen & Mon)

---

---

### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:HISTory?

---

<b>Description</b>	<p>This query returns the history status of stream errors analysis.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; QoS Metrics</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:HISTory? &lt;wsp&gt;OUTSequence   FLOSs</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OUTSequence   FLOSs</p> <p>Selects the type of stream error analysis error.</p> <p>OUTSequence: Out of Sequence</p> <p>FLOSs: Frame Loss</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of FLOS   OUTS error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:STR:HIST? FLOS</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:CURRent?</p>

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:CURRent?**

---

<b>Description</b>	<p>This query returns the current status of stream errors analysis.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; QoS Metrics</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:CURRent? &lt;wsp&gt;OUTSequence   FLOSs</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OUTSequence   FLOSs</p> <p>Selects the type of stream error analysis error.</p> <p>OUTSequence: Out of Sequence</p> <p>FLOSs: Frame Loss</p>
<b>Response Syntax</b>	<code>&lt;Current&gt;</code>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of FLOS   OUTS error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:ETH:ERR:STR:CURR? FLOS</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:STReam:HISTory?</code>

---

# Summary (RFC 6349)

---

:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:WINDow?

---

<b>Description</b>	<p>This query returns the TCP Throughput Total Max Window.</p> <p>At *RST, this value is device dependent.</p> <p>Navigation Path: Results &gt; Summary &gt; TCP Throughput &gt; Window</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:WINDow? &lt;wsp&gt;LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Max Window&gt;</p>
<b>Response(s)</b>	<p><b>Max Window:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Total Max Window</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:TCP:THR:WINDOW? RTOL</p> <p>Returns TCP Throuput Total Max Window for remote to local.</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:BUFFer:DELAy?</p>

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:ACTUal:L?**

<b>Description</b>	<p>This query returns the average of actual TCP Throughput metric.</p> <p>At *RST, this value is device dependent.</p> <p>Navigation Path: Results &gt; Summary &gt; TCP Throughput &gt; Actual L4</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:ACTUal:L? <wsp>LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Actual L>
<b>Response(s)</b>	<p><b>Actual L:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the average of actual TCP Throughput metric.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:TCP:THR:ACTU:L? LTOR</p> <p>Returns the Actual L value for local to remote.</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:WINDow?

## SCPI Command Reference

Summary (RFC 6349)

---

---

### :FETCh[1..n]:DATA:TELeom:ETHernet:RFC:TCP:THRoughput:IDEal:L?

---

<b>Description</b>	<p>This query returns the ideal TCP throughput metric.</p> <p>At *RST, this value is device dependent.</p> <p>Navigation Path: Results &gt; Summary &gt; TCP Throughput &gt; Ideal L4</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:ETHernet:RFC:TCP:THRoughput:IDEal:L? <wsp>LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Ideal L>
<b>Response(s)</b>	<p><b>Ideal L:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Ideal L</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:TCP:THR:IDE:L? LTOR</p> <p>Returns the Ideal L for local to remote.</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:ETHernet:RFC:TCP:THRoughput:IDEal:L?

---

**:FETCh[1..n]:DATA:TELeom:ETHernet:RFC:TCP:THRoughput:TCP:EFFiciency?**

<b>Description</b>	<p>This query returns the TCP Efficiency metric.</p> <p>At *RST, this value is device dependent.</p> <p>Navigation Path: Results &gt; Summary &gt; TCP Throughput &gt; TCP Efficiency</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELeom:ETHernet:RFC:TCP:THRoughput:TCP:EFFiciency? &lt;wsp&gt;LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;TCP Efficiency&gt;</p>
<b>Response(s)</b>	<p><b>TCP Efficiency:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the TCP Efficiency metric</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:TCP:THR:TCP:EFF? LTOR</p> <p>Returns TCP Efficiency metric value for local to remote.</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELeom:ETHernet:RFC:TCP:THRoughput:THReshold?</p>

## SCPI Command Reference

Summary (RFC 6349)

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:BUFFer:DELAy?**

---

<b>Description</b>	<p>This query returns the ideal Buffer Delay metric.</p> <p>At *RST, this value is device dependent.</p> <p>Navigation Path: Results &gt; Summary &gt; TCP Throughput &gt; Buffer Delay</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:BUFFer:DELAy? &lt;wsp&gt;LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Buffer delay&gt;</p>
<b>Response(s)</b>	<p><b>Buffer delay:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Buffer Delay</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:TCP:THR:BUFF:DEL? LTOR</p> <p>Returns ideal Buffer Delay metric for local to remote.</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:TCP:EFFiciency?</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:L:ACTUal?**

<b>Description</b>	<p>This query returns the each step average TCP throughput metric</p> <p>At *RST, this value is device dependent.</p> <p>Navigation Path: Results &gt; Summary &gt; Window Sweep &gt; Actual L4</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:WINDow:SWEep:L:ACTUal? <wsp> <Step>, LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Step:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the step from 1 to 4</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Sweep L>
<b>Response(s)</b>	<p><b>Sweep L:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Window sweep actual L.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:WIND:SWE:ACTU:L? 1,LTOR</p> <p>Returns the average TCP throughput metric for step1 local to remote.</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:ACTUal:L?

## SCPI Command Reference

Summary (RFC 6349)

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:TCP:THRoughput:ACTUal:L:VERDict?**

<b>Description</b>	<p>This query returns the average of actual TCP Throughput metric verdict</p> <p>At *RST, this value is device dependent.</p> <p>Navigation Path: Results &gt; Summary &gt; TCP Throughput &gt; Actual L4 verdict.</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:TCP:THRoughput:ACTUal:L:VERDict? &lt;wsp&gt;LTOR   RTOL</p>
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Verdict&gt;</p>
<b>Response(s)</b>	<p><b>Verdict:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the TCP Throughput metric verdict.</p> <p>PASS, Verict is Pass.</p> <p>FAIL, Verdict is Fail.</p> <p>UNDEFINED, Undefined verdict.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:TCP:THR:ACTU:L:VERD? LTOR</p> <p>Returns the actual TCP Throughput metric verdict for local to remote.</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:ETHernet:RFC:TCP:THRoughput:ACTUal:L?</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold?**

<b>Description</b>	<p>This query returns the average of actual TCP Throughput threshold.</p> <p>At *RST, this value is device dependent.</p> <p>Navigation Path: Results &gt; Summary &gt; TCP Throughput &gt; TCP Throughput threshold</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:THReshold? <wsp>LTOR   RTOL
<b>Parameter(s)</b>	<p><b>Direction:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Threshold>
<b>Response(s)</b>	<p><b>Threshold:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the average of actual TCP Throughput threshold</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:RFC:TCP:THR:THR? LTOR</p> <p>Returns TCP Throughput threshold for local to remote.</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:TCP:THRoughput:BUFFer:DELAy?

## SCPI Command Reference

Summary (RFC 6349)

---

:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:MTU?

<b>Description</b>	This query returns the Maximum Transfer Unit At *RST, this value is device dependent. Navigation Path: Results > Summary > MTU (Bytes)
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:MTU?
<b>Response Syntax</b>	<MTU>
<b>Response(s)</b>	<b>MTU:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the Maximum Transfer Unit.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:MTU? Returns the Maximum Transfer Unit
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:MINimum:RTT?

---



**:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:MINimum:RTT?**

<b>Description</b>	This query returns the Minimum RTT. At *RST, this value is device dependent. Navigation Path: Results > Summary > Minimum RTT(ms)
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:RFC:MINimum:RTT?
<b>Response Syntax</b>	<Minimum RTT>
<b>Response(s)</b>	<b>Minimum RTT:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the Minimum RTT.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:RFC:MIN:RTT? Returns the Minimum RTT
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:RFC:MTU?

---

## Alarms/Errors

**:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the pattern alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER Alarm</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; BER Alarm</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:SEConds? &lt;wsp&gt;PLOSs   NTRaffic</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PLOSs   NTRaffic</p> <p>Selects the number of seconds within which the pattern alarm occurred.</p> <p>PLOSs: Pattern Loss.</p> <p>NTRaffic: No Traffic.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds within which the pattern alarm occurred.</p> <p>PLOSs, selects the type of pattern alarm as Pattern Loss.</p> <p>NTRaffic, selects the type of pattern alarm as No Traffic.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:PATT:ALAR:PATT:SEC? PLOS</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:SEConds?</p>

**:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:HISTory?**

<b>Description</b>	<p>This query returns the history status of the pattern alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER Alarm</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; BER Alarm</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:HISTory? <wsp>PLOSs   NTRaffic
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PLOSs   NTRaffic</p> <p>Selects the history status of the pattern alarm.</p> <p>PLOSs: Pattern Loss.</p> <p>NTRaffic: No Traffic.</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of the pattern alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PATT:ALAR:PATT:HIST? PLOS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:PATtern:ALARm:PATtern:CURRent?

---

<b>Description</b>	<p>This query returns the current status of the pattern alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER Alarm</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; BER Alarm</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:PATtern:ALARm:PATtern:CURRent? <wsp>PLOSs   NTRaffic
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PLOSs   NTRaffic</p> <p>Selects the current status of the pattern alarm.</p> <p>PLOSs: Pattern Loss.</p> <p>NTRaffic: No Traffic.</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of the pattern alarm.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PATT:ALAR:PATT:CURR? PLOS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:PATtern:ALARm:PATtern:HISTory?

---

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the pattern error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER Error</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; BER Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:SEConds? <wsp>BIT   MISMATCH0   MISMATCH1
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT   MISMATCH0   MISMATCH1</p> <p>Selects the number of seconds within which pattern error occurred.</p> <p>BIT: Bit.</p> <p>MISMATCH0: Mismatch '0'.</p> <p>MISMATCH1: Mismatch '1'.</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds within which the pattern error occurred.</p> <p>BIT, selects the type of pattern error as Bit.</p> <p>MISMATCH0, selects the type of pattern error as Mismatch '0'.</p> <p>MISMATCH1, selects the type of pattern error as Mismatch '1'.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:AMO 25</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:INJ</p> <p>FETC:DATA:TEL:PATT:ERR:PATT:SEC? BIT</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:SEConds?

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELecom:PATtern:ERRor:PATtern:COUNT?

<b>Description</b>	<p>This query returns the count of pattern error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER Error</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; BER Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:PATtern:ERRor:PATtern:COUNT? <wsp>BIT   MISMATCH0   MISMATCH1
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT   MISMATCH0   MISMATCH1</p> <p>Selects the count of pattern error.</p> <p>BIT: Bit.</p> <p>MISMATCH0: Mismatch '0'.</p> <p>MISMATCH1: Mismatch '1'.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of pattern error.</p> <p>BIT, selects the type of pattern error as Bit.</p> <p>MISMATCH0, selects the type of pattern error as Mismatch '0'.</p> <p>MISMATCH1, selects the type of pattern error as Mismatch '1'.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:AMO 25</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:INJ</p> <p>FETC:DATA:TEL:PATT:ERR:PATT:COUN? BIT</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:PATtern:ERRor:PATtern:HISTory?

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:RATE?**

<b>Description</b>	<p>This query returns the current rate of pattern error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER Error</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; BER Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:RATE? <wsp>BIT   MISMATCH0   MISMATCH1
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT   MISMATCH0   MISMATCH1</p> <p>Selects the current rate of pattern error.</p> <p>BIT: Bit.</p> <p>MISMATCH0: Mismatch '0'.</p> <p>MISMATCH1: Mismatch '1'.</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of pattern error.</p> <p>BIT, selects the type of pattern error as Bit.</p> <p>MISMATCH0, selects the type of pattern error as Mismatch '0'.</p> <p>MISMATCH1, selects the type of pattern error as Mismatch '1'.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:AMO 25</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:INJ</p> <p>FETC:DATA:TEL:PATT:ERR:PATT:RATE? BIT</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:PATTern:ERRor:PATTern:AMOunt

---

<b>Description</b>	<p>This command sets the amount of pattern error to inject into the instrument.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; Bit Error</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (BER) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:PATTern:ERRor:PATTern:AMOunt &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT SOUR:DATA:TEL:PATT:ERR:PATT:AMO 25 Returns: 25</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:PATTern:ERRor:PATTern:INJect</pre>

---



**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AMOUt?**

<b>Description</b>	<p>This query returns the amount of pattern error injected into the instrument.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; Bit Error</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (BER) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AMOUt?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of pattern error to inject into the instrument is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:AMO?</p> <p>Returns: 25</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:INJect

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE**

<b>Description</b>	<p>This command sets the manual type of pattern error.</p> <p>At *RST condition, this value is set to BIT.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; Bit Error</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE &lt;wsp&gt;BIT</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT</p> <p>Selects the manual type pattern error.</p> <p>BIT: Bit</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE?</p> <p>Returns: BIT</p>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE</code>

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type of pattern error.</p> <p>At *RST condition, this value is set to BIT.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; Bit Error</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the manual type pattern error.</p> <p>BIT, bit is selected as the manual type pattern error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE?</p> <p>Returns: BIT</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE?

---

## SCPI Command Reference

### *Alarms/Errors*

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:INJect**

<b>Description</b>	<p>This command injects the pattern error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; Bit Error</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; Bit Error</p> <p>Navigation Path: Test &gt; BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:INJect
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:INJ</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE?

---

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CHAN**

---

<b>Description</b>	<p>This command selects the channel used for pattern error injection in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER &gt; Errors</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CHAN &lt;wsp&gt; &lt;Channel&gt;</code>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel used for pattern error injection in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:PATT:ERR:PATT:CHAN 5</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CHANnel?</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CHAN?**

<b>Description</b>	<p>This query returns the channel used for pattern error injection in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER &gt; Errors</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CHAN?</p>
<b>Response Syntax</b>	<p>&lt;Channel&gt;</p>
<b>Response(s)</b>	<p><b>Channel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the channel used for pattern error injection in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:CHAN 3</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:CHAN?</p> <p>Returns 3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CHANnel</p>

---

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:HISTory?**

<b>Description</b>	<p>This query returns the history status of pattern error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER Error</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; BER Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:HISTory? <wsp>BIT   MISMATCH0   MISMATCH1
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT   MISMATCH0   MISMATCH1</p> <p>Selects the history type of pattern error.</p> <p>BIT: Bit</p> <p>MISMATCH0: Mismatch '0'</p> <p>MISMATCH1: Mismatch '1'</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of pattern error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:AMO 25</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:INJ</p> <p>FETC:DATA:TEL:PATT:ERR:PATT:HIST? BIT</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:COUNt?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CURRent?

---

<b>Description</b>	<p>This query returns the current status of pattern error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER Error</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; BER Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:CURRent? <wsp>BIT   MISMATCH0   MISMATCH1
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT   MISMATCH0   MISMATCH1</p> <p>Selects the current type of pattern error.</p> <p>BIT: Bit</p> <p>MISMATCH0: Mismatch '0'</p> <p>MISMATCH1: Mismatch '1'</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of pattern error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:AMO 25</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:INJ</p> <p>FETC:DATA:TEL:PATT:ERR:PATT:CURR? BIT</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:RATE?

---



---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE**

---

<b>Description</b>	<p>This command sets the automated type of pattern error.</p> <p>At *RST condition, this value is set to BIT.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; Bit Error</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE <wsp>BIT
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BIT</p> <p>Selects the AUTOMATED type pattern error.</p> <p>BIT: Bit</p>
<b>Example(s)</b>	SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE BIT
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

<b>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE?</b>	
<b>Description</b>	<p>This query returns the automated type of pattern error.</p> <p>At *RST condition, this value is set to BIT.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER Error</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; BER Error</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the automated type pattern error.</p> <p>BIT, bit is selected as automated type pattern error.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:MANual:TYPE

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:RATE**

<b>Description</b>	<p>This command sets the rate of pattern error to inject.</p> <p>At *RST condition, this value is set to 1.0E-04.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER &gt; Rate</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the rate of error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:PATT:ERR:PATT:AUT:RATE 1.0E-03 SOUR:DATA:TEL:PATT:ERR:PATT:AUT:RATE? Returns: 1.0E-03</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AMount</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:RATE?

---

<b>Description</b>	<p>This query returns the rate of pattern error to inject.</p> <p>At *RST condition, this value is set to 1.0E-04.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER &gt; Rate</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (BERT) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:RATE?[ &lt;wsp&gt;MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current rate of pattern error to inject into the instrument is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Error&gt;</code>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of pattern error to inject into the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:AUT:RATE 1.0E-03</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:AUT:RATE?</p> <p>Returns: 1.0E-03</p>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AMount?</code>

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:CONTInuous**

<b>Description</b>	<p>This command sets the continuous pattern error to inject into the instrument.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER&gt; Bit Error</p> <p>Navigation Path: Test &gt; BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (BER) &gt; Type (Errors) &gt; Mode</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the continuous pattern error to inject into the instrument.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:PATT:ERR:PATT:AUT:CONT OFF SOUR:DATA:TEL:PATT:ERR:PATT:AUT:CONT? Returns: 0</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:CONTinuous?**

<b>Description</b>	<p>This query returns the continuous pattern error to inject into the instrument.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER &gt; Bit Error</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (BER) &gt; Type (Errors) &gt; Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:CONTinuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the continuous mode pattern error status.</p> <p>1, sets the continuous pattern error to ON.</p> <p>0, sets the continuous pattern error to OFF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:AUT:CONT OFF</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:AUT:CONT?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated**

<b>Description</b>	<p>This command sets the automated pattern type.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the automated pattern type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:AUT OFF</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:AUT?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:CONTinuous

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated?

---

<b>Description</b>	<p>This query returns the automated pattern type.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the automated pattern status.</p> <p>1, sets the automated pattern type to ON.</p> <p>0, sets the automated pattern type to OFF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ERR:PATT:AUT OFF</p> <p>SOUR:DATA:TEL:PATT:ERR:PATT:AUT?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:CONTInuous?

---



---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:TYPE?**

---

<b>Description</b>	<p>This query returns the pattern alarm type.</p> <p>At *RST condition, this value is set to Pattern Loss.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; BERT Alarm</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of pattern alarm.</p> <p>PLOSs, Pattern Loss is selected as type of pattern alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE PLOS</p> <p>SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE?</p> <p>Returns: PLOS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern</p>

---

## SCPI Command Reference

### *Alarms/Errors*

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CHAN**

<b>Description</b>	<p>This command selects the channel used for pattern alarm injection in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER &gt; Alarms</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CHAN &lt;wsp&gt; &lt;Channel&gt;</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel used for pattern alarm injection in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ALAR:PATT:CHAN 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CHANnel?</p>

---

**:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CHAN?**

<b>Description</b>	<p>This query returns the channel used for pattern error injection in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; BERT &gt; Results &gt; Summary &gt; BER &gt; Alarms</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CHAN?
<b>Response Syntax</b>	<Channel>
<b>Response(s)</b>	<p><b>Channel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the channel used for pattern alarm injection in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PATT:ALAR:PATT:CHAN 3</p> <p>SOUR:DATA:TEL:PATT:ALAR:PATT:CHAN?</p> <p>Returns 3</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:CHANnel

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :INPut[1..n]:TELeom:BACKplane:ALARm:STATus:HISTory?

---

<b>Description</b>	<p>This query returns the history status of the clock alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarm/Errors &gt; Clock</p>
<b>Syntax</b>	:INPut[1..n]:TELeom:BACKplane:ALARm:STATus:HISTory?
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of the LOC Alarm</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	INP:TEL:BACK:ALAR:STAT:HIST?
<b>See Also</b>	INPut[1..n]:TELeom:BACKplane:ALARm:STATus:SECOnds?

---

---

**:INPut[1..n]:TELeom:BACKplane:ALARm:STATus:CURRent?**

---

<b>Description</b>	<p>This query returns the current status of the clock alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarm/Errors &gt; Clock</p>
<b>Syntax</b>	:INPut[1..n]:TELeom:BACKplane:ALARm:STATus:CURRent?
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of LOC Alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	INP:TEL:BACK:ALAR:STAT:CURR?
<b>See Also</b>	INPut[1..n]:TELeom:BACKplane:ALARm:STATus:HISTory?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :INPut[1..n]:TELecom:BACKplane:ALARm:STATus:SEConds?

---

<b>Description</b>	This query returns the number of seconds for which the alarm of the clock alarm was raised. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Alarm/Errors > Clock
<b>Syntax</b>	:INPut[1..n]:TELecom:BACKplane:ALARm:STATus:SEConds?
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<b>Seconds:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds the alarm was raised.
<b>Example(s)</b>	INP:TEL:BACK:ALAR:STAT:SEC?
<b>See Also</b>	INPut[1..n]:TELecom:BACKplane:ALARm:STATus:CURRent?

---

**:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:HISTory?**

<b>Description</b>	<p>This query returns the history status of an optical port alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE/Serial(OTU3/OTU3E1/OTU3E2)/OC-768/STM-256&gt;Results &gt; Alarm/Errors &gt; Interface</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:HISTory? <wsp>LOS
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOS</p> <p>Selects the history status of an optical port alarm.</p> <p>LOS</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of an optical port alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OPT:ALAR:PORT:GLOB:HIST? LOS
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:HISTory?</p> <p>FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:CURREnt?</p> <p>FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:SECConds?</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:CURRent?

---

<b>Description</b>	<p>This query returns the current status of an optical port alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE/Serial(OTU3/OTU3E1/OTU3E2)/OC-768/STM-256&gt;Results &gt; Alarm/Errors &gt; Interface</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:CURRent? <wsp>LOS
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOS</p> <p>Selects the current status of port alarm.</p> <p>LOS</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of Port alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OPT:ALAR:PORT:GLOB:CURR? LOS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:HISTory? FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:CURRent? FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:SEConds?

---



**:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:HISTory?**

<b>Description</b>	<p>This query returns the history status of global CAUI Lanes alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE/Serial(OTU3/OTU3E1/OTU3E2)/STM-256/OC-768 &gt; Results &gt; Alarm/Errors &gt; Interface</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:HISTory? <wsp>FREQUENCY   LOC   INTerlinkstatus   SERialfreq
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FREQUENCY   LOC   INTerlinkstatus   SERialfreq</p> <p>Selects the history status of global CAUI Lanes alarm.</p> <p>FREQUENCY: FREQUENCY</p> <p>LOC: Loss of Clock</p> <p>INTerlinkstatus: INTerlinkstatus</p> <p>SERialfreq: SERialfreq</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of CAUI alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:CAUI:ALAR:GLOB:HIST? FREQ
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:CURREnt?**

<b>Description</b>	<p>This query returns the current status of global CAUI Lanes alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE/Serial(OTU3/OTU3E1/OTU3E2)/STM-256/OC-768 &gt; Results &gt; Alarm/Errors &gt; Interface</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:CURREnt? &lt;wsp&gt;FREQUENCY   LOC   INTErlinkstatus   SERIalfreq</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FREQUENCY   LOC   INTErlinkstatus   SERIalfreq</p> <p>Selects the current status of global CAUI Lanes alarm.</p> <p>FREQUENCY: FREQUENCY</p> <p>LOC: Loss of Clock</p> <p>INTErlinkstatus: INTErlinkstatus</p> <p>SERIalfreq: SERIalfreq</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Current status of CAUI alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:CAUI:ALAR:GLOB:CURR? FREQ</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:HISTory?</p>

**:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:RX:HIStory?**

<b>Description</b>	<p>This query returns the history status of an Optical Lanes alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarm/Errors &gt; Interface &gt; Interface</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:RX:HIStory? <wsp> <Lane>, LOS
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the history status of an Optical alarm. The range for the Optical lane is from 0 to 9.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOS</p> <p>Sets the per lane history status of the type of port alarm LOS for optical lanes.</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of Port alarm for per lane configuration.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OPT:ALAR:RX:HIST? 1, LOS
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:HISTory?</p> <p>FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:CURRent?</p> <p>FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:SECOnds?</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:RX:CURrent?

---

<b>Description</b>	<p>This query returns the current status of an Optical Lanes alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarm/Errors &gt; Interface &gt; Interface</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:RX:CURrent? <wsp> <Lane>, LOS
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the current status of an Optical alarm. The range for the Optical lane is from 0 to 9.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOS</p> <p>Sets the per lane current status of the type of port alarm LOS for optical lanes.</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of Port alarm for per lane configuration.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OPT:ALAR:RX:CURR? 1, LOS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:HISTory? FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:CURRent? FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:SECOnds?

---

**:FETCh[1..n]:DATA:TELEcom:OPTical:ALArm:RX:SEConds?**

<b>Description</b>	<p>This query returns the seconds value of an Optical Lanes alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarm/Errors &gt; Interface &gt; Interface</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OPTical:ALArm:RX:SEConds? <wsp> <Lane>, LOS
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the seconds value of an Optical alarm. The range for the Optical lane is from 0 to 9.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOS</p> <p>Sets the per lane seconds value of the type of port alarm LOS for optical lanes.</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the seconds value of Port alarm for per lane configuration.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OPT:ALAR:RX:SEC? 1, LOS
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:HISTory?</p> <p>FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:CURRent?</p> <p>FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:SEConds?</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:HISTory?

---

<b>Description</b>	<p>This query returns the history status of CAUI Lanes alarm for the selected traffic stream.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarm/Errors &gt; Interface</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:HISTory? &lt;wsp&gt; &lt;Lane&gt;, FREquency   LOC</p>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the history status of an CAUI alarm. The range for the CAUI lane is from 0 to 9.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FREquency   LOC</p> <p>Selects the history status of global CAUI Lanes alarm.</p> <p>FREQuency: Frequency</p> <p>LOC: Loss of Clock</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of CAUI alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:CAUI:ALAR:LANE:HIST? 1,LOC</p> <p>FETC:DATA:TEL:CAUI:ALAR:LANE:HIST? 1,FREQ</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBAL:HISTory?</p>

---

**:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:CURRent?**

<b>Description</b>	<p>This query returns the current status of CAUI Lanes alarm for the selected traffic stream.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarm/Errors &gt; Interface</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:CURRent? <wsp> <Lane>, FREquency   LOC
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the current status of an CAUI alarm. The range for the CAUI lane is from 0 to 9.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FREquency   LOC</p> <p>Selects the type of port alarm.</p> <p>FREQuency: Frequency</p> <p>LOC: Loss of Clock</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Current status of CAUI alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:CAUI:ALAR:LANE:CURR? 1,LOC</p> <p>FETC:DATA:TEL:CAUI:ALAR:LANE:CURR? 1,FREQ</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:SEConds?

---

<b>Description</b>	<p>This query returns the seconds value of CAUI Lanes alarm for the selected traffic stream.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarm/Errors &gt; Interface</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:LANE:SEConds? &lt;wsp&gt; &lt;Lane&gt;, FREquency   LOC</p>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the seconds value of global CAUI Lanes alarm. The range for the CAUI lane is from 0 to 9.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FREquency   LOC</p> <p>Selects the type of port alarm.</p> <p>FREQuency: Frequency.</p> <p>LOC: Loss of Clock</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the seconds value of CAUI alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:CAUI:ALAR:LANE:SEC? 1,LOC</p> <p>FETC:DATA:TEL:CAUI:ALAR:LANE:SEC? 1,FREQ</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBAL:SEConds?</p>

---



**:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:TYPE?**

<b>Description</b>	<p>This query returns port alarm type.</p> <p>At *RST condition, this value is set to LOS.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Defect Summary &gt; Layer (Interface) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of port alarm.</p> <p>LOS, Loss of Signal (LOS) is selected as type of port alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OPT:ALAR:PORT:TYPE LOS</p> <p>SOUR:DATA:TEL:OPT:ALAR:PORT:TYPE?</p> <p>Returns: LOS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT

---

<b>Description</b>	<p>This command enables or disables the status of the connector alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Interface &gt; Alarms</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the connector alarm generation.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OPT:ALAR:PORT ON</p> <p>SOUR:DATA:TEL:OPT:ALAR:PORT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical?</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT?**

<b>Description</b>	<p>This query returns the status of connector alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Interface &gt; Alarms</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of connector alarm generation.</p> <p>1, alarm generation is enabled.</p> <p>0, alarm generation is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OPT:ALAR:PORT ON</p> <p>SOUR:DATA:TEL:OPT:ALAR:PORT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE**

<b>Description</b>	<p>This command sets the particular lane number for the injection purpose.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Interface &gt; Alarm</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE &lt;wsp&gt; &lt;Lane&gt;, ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which injection is to be done for connector alarm.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the per lane status for connector alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OPT:ALAR:PORT:LANE 1, ON</p> <p>SOUR:DATA:TEL:OPT:ALAR:PORT:LANE? 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALANes</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:ALANes</p>

**:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE?**

<b>Description</b>	<p>This query returns the particular lane number for the injection purpose.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Interface &gt; Alarm</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which injection is to be done for connector alarm.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of connector alarm lane.</p> <p>1, particular lane is enabled</p> <p>0, particular lane is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OPT:ALAR:PORT:LANE 1, ON</p> <p>SOUR:DATA:TEL:OPT:ALAR:PORT:LANE? 1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALANes?</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:ALANes</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:ALANes

---

<b>Description</b>	<p>This command sets all lanes for the injection purpose.</p> <p>This command is not associated with any *RST condition.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Interface &gt; Alarm</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:ALANes <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the all lane status for connector alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OPT:ALAR:PORT:ALAN ON</p> <p>SOUR:DATA:TEL:OPT:ALAR:PORT:ALAN?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTL:LANE</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE</p>

---

**:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:ALANes?**

<b>Description</b>	<p>This query returns all lanes for the injection purpose.</p> <p>This command is not associated with any *RST condition.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Interface &gt; Alarm</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:ALANes?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of connector alarm for all lanes.</p> <p>1, all lanes are enabled.</p> <p>0, all lanes are disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OPT:ALAR:PORT:ALAN ON</p> <p>SOUR:DATA:TEL:OPT:ALAR:PORT:ALAN?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTL:LANE ?</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:LANE</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:HISTory?

---

<b>Description</b>	<p>This query returns the history status of the higher layer User Datagram Protocol (UDP) error. At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; IP/UDP &gt; UDP Checksum</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:HISTory? &lt;wsp&gt;   UDPChecksum  </p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   UDPChecksum  </p> <p>Selects the higher layer UDP (User Data Protocol) type of protocol error.</p> <p>UDPChecksum: UDP Checksum.</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of higher layer UDP type protocol error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:UDPP:HIST? UDPC</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:HISTory?</p>

---



**:FETCh[1..n]:DATA:TELeom:ETHernet:ERRor:UDPProtocol:CURRent?**

<b>Description</b>	This query returns the current status of the higher layer User Datagram Protocol (UDP) error. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Alarms/Errors > IP/TCP/UDP > UDP Checksum
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:ETHernet:ERRor:UDPProtocol:CURRent? <wsp>   UDPChecksum
<b>Parameter(s)</b>	<b>Error:</b> The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are:   UDPChecksum   Selects the higher layer UDP (User Datagram Protocol) type of protocol error. UDPChecksum: UDP Checksum.
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of higher layer UDP type protocol error. PRESENT, indicates that at least one error has occurred in the last second. ABSENT, indicates that there is no error. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:UDPP:CURR? UDPC
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:ETHernet:ERRor:IPPRotocol:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which higher layer User Datagram Protocol (UDP) error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; IP/TCP/UDP &gt; UDP Checksum</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:SEConds? <wsp>   UDPChecksum
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   UDPChecksum  </p> <p>Selects the higher layer UDP (User Datagram Protocol) type of protocol error.</p> <p>UDPChecksum: UDP Checksum.</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of higher layer UDP type protocol error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:UDPP:SEC? UDPC
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:SEConds?

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:UDPProtocol:COUNT?**

<b>Description</b>	<p>This query returns the count of higher layer User Datagram Protocol (UDP) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; IP/TCP/UDP &gt; UDP Checksum</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:UDPProtocol:COUNT? &lt;wsp&gt;   UDPChecksum  </code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   UDPChecksum  </p> <p>Selects the higher layer UDP (User Datagram Protocol) type of protocol error.</p> <p>UDPChecksum: UDP Checksum.</p>
<b>Response Syntax</b>	<code>&lt;Count&gt;</code>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of higher layer UDP type protocol error.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:ETH:ERR:UDPP:COUN? UDPC</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:IPPRotocol:COUNT?</code>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:RATE?

---

<b>Description</b>	<p>This query returns the current rate of the higher layer User Datagram Protocol (UDP) error. At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; IP/TCP/UDP &gt; UDP Checksum</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:RATE? <wsp>   UDPChecksum
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   UDPChecksum  </p> <p>Selects the higher layer UDP (User Datagram Protocol) type of protocol error.</p> <p>UDPChecksum: UDP Checksum.</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of higher layer UDP type protocol error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:UDPP:RATE? UDPC
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:RATE?

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:HISTory?**

---

<b>Description</b>	<p>This query returns the history status of the higher layer Internet Protocol (IP) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; IP/TCP/UDP &gt; IP Checksum</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:HISTory? &lt;wsp&gt;   IPChecksum  </code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   IPChecksum  </p> <p>Selects the type of higher layer IP (Internet Protocol) error.</p> <p>IPChecksum: IP Header Checksum.</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of higher layer IP type protocol error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:ETH:ERR:IPPR:HIST? IPCH</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:HISTory?</code>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:IPPRotocol:CURRent?

---

<b>Description</b>	<p>This query returns the current status of the higher layer Internet Protocol (IP) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; IP/TCP/UDP &gt; IP Checksum</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:IPPRotocol:CURRent? &lt;wsp&gt;   IPChecksum  </p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   IPChecksum  </p> <p>Selects the type of higher layer IP (Internet Protocol) error.</p> <p>IPChecksum: IP Header Checksum.</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of higher layer IP type protocol error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:IPPR:CURR? IPCH</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:UDPProtocol:CURRent?</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which higher layer Internet Protocol (IP) error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; IP/TCP/UDP &gt; IP Checksum</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:SEConds? &lt;wsp&gt;   IPChecksum  </code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   IPChecksum  </p> <p>Selects the type of higher layer IP (Internet Protocol) error.</p> <p>IPChecksum: IP Header Checksum.</p>
<b>Response Syntax</b>	<code>&lt;Seconds&gt;</code>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of higher layer IP type protocol error.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:ETH:ERR:IPPR:SEC? IPCH</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:SEConds?</code>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:COUNT?

---

<b>Description</b>	<p>This query returns the count of the higher layer IP protocol error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; IP/TCP/UDP &gt; IP Checksum</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:COUNT? &lt;wsp&gt;   IPChecksum  </p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   IPChecksum  </p> <p>Selects the type of higher layer IP (Internet Protocol) error.</p> <p>IPChecksum: IP Header Checksum.</p>
<b>Response Syntax</b>	<p>&lt;Count&gt;</p>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of higher layer IP type protocol error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:IPPR:COUN? IPCH</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:COUNT?</p>

---



---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:RATE?**

---

<b>Description</b>	<p>This query returns the current rate of the higher layer IP protocol error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; IP/TCP/UDP &gt; IP Checksum</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:IPPRotocol:RATE? &lt;wsp&gt;   IPChecksum</code> 
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are:   IPChecksum  </p> <p>Selects the type of higher layer IP (Internet Protocol) error.</p> <p>IPChecksum: IP Header Checksum.</p>
<b>Response Syntax</b>	<code>&lt;Rate&gt;</code>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of higher layer IP type protocol error.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:ETH:ERR:IPPR:RATE? IPCH</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:UDPProtocol:RATE?</code>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:HISTory?

---

<b>Description</b>	<p>This query returns the history status of MAC error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; FCS/Jabber/Runt/Oversize/Undersize</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:HISTory? &lt;wsp&gt;FCS   JABBer   OVERsize   RUNT   UNDersize</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCS   JABBer   OVERsize   RUNT   UNDersize</p> <p>Selects the type of MAC (Media Access Control) error.</p> <p>FCS: Frame Check Sequence.</p> <p>JABBer: Jabber/Giant.</p> <p>OVERsize: Oversize.</p> <p>RUNT: Runt.</p> <p>UNDersize: Undersize.</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of MAC error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:MAC:HIST? JABB</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:CURRent?</p>

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:CURRent?**

---

<b>Description</b>	<p>This query returns the current status of MAC error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; FCS/Jabber/Runt/Oversize/Undersize</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:CURRent? &lt;wsp&gt;FCS   JABBer   OVERsize   RUNT   UNDersize</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCS   JABBer   OVERsize   RUNT   UNDersize</p> <p>Selects the type of MAC (Media Access Control) error.</p> <p>FCS: Frame Check Sequence.</p> <p>JABBer: Jabber/Giant.</p> <p>OVERsize: Oversize.</p> <p>RUNT: Runt.</p> <p>UNDersize: Undersize.</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of MAC error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:MAC:CURR? JABB</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:SECOnds?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which MAC error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; FCS/Jabber/Runt/Oversize/Undersize</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:SEConds? &lt;wsp&gt;FCS   JABBer   OVERsize   RUNT   UNDersize</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCS   JABBer   OVERsize   RUNT   UNDersize</p> <p>Selects the type of MAC (Media Access Control) error.</p> <p>FCS: Frame Check Sequence.</p> <p>JABBer: Jabber/Giant.</p> <p>OVERsize: Oversize.</p> <p>RUNT: Runt.</p> <p>UNDersize: Undersize.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the number of seconds of MAC error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:MAC:SEC? JABB</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:COUNt?</p>

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:COUNT?**

<b>Description</b>	<p>This query returns the count of MAC error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; FCS/Jabber/Runt/Oversize/Undersize</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:COUNT? <wsp>FCS   JABBer   OVERsize   RUNT   UNDersize
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCS   JABBer   OVERsize   RUNT   UNDersize</p> <p>Selects the type of MAC (Media Access Control) error.</p> <p>FCS: Frame Check Sequence.</p> <p>JABBer: Jabber/Giant.</p> <p>OVERsize: Oversize.</p> <p>RUNT: Runt.</p> <p>UNDersize: Undersize.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the count of MAC error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:MAC:COUN? JABB
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:RATE?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:RATE?

---

<b>Description</b>	<p>This query returns the current rate of MAC error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; FCS/Jabber/Runt/Oversize/Undersize</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:RATE? &lt;wsp&gt;FCS   JABBer   OVERsize   RUNT   UNDersize</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCS   JABBer   OVERsize   RUNT   UNDersize</p> <p>Selects the type of MAC (Media Access Control) error.</p> <p>FCS: Frame Check Sequence.</p> <p>JABBer: Jabber/Giant.</p> <p>OVERsize: Oversize.</p> <p>RUNT: Runt.</p> <p>UNDersize: Undersize.</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the current rate of MAC error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:MAC:RATE? JABB</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:HISTory?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:OVERsize**

---

<b>Description</b>	<p>This command allows monitoring of the Oversize frame errors.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Oversized Monitoring</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:OVERsize <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Allows monitoring of the Oversize frame errors.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:MAC:OVER ON</p> <p>SOUR:DATA:TEL:ETH:ERR:MAC:OVER?</p> <p>Returns: 1</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:CURRent?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:OVERsize?

---

<b>Description</b>	<p>This query returns the status of the monitoring of the Oversize frame errors.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Oversized Monitoring</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:OVERsize?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the monitoring of Oversize frame errors.</p> <p>1, oversize frame error is ON.</p> <p>0, oversize frame error is OFF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:MAC:OVER ON</p> <p>SOUR:DATA:TEL:ETH:ERR:MAC:OVER?</p> <p>Returns: 1</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:MAC:CURRent?

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type Media Access Control (MAC) error.</p> <p>At *RST condition, this value is set to FCS.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the manual type Media Access Control (MAC) error.</p> <p>FCS, Frame Check Sequence (FCS) as manual type MAC error is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:MAC:MAN:TYPE?</p> <p>Returns: FCS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJECT</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOut

---

<b>Description</b>	<p>This command sets the amount of Media Access Control (MAC) error to inject into the instrument.</p> <p>At *RST condition, this value is set to MINimum.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOut <wsp> <Amount>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:MAC:AMO 50</p> <p>SOUR:DATA:TEL:ETH:ERR:MAC:AMO?</p> <p>Returns: 50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOut?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJect</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUnt?**

<b>Description</b>	<p>This query returns the amount of MAC error to inject.</p> <p>At *RST condition, this value is set to MINimum.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUnt?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of MAC error to inject into the instrument is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of MAC error to inject into the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:MAC:AMO 50</p> <p>SOUR:DATA:TEL:ETH:ERR:MAC:AMO?</p> <p>Returns: 50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUnt</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJect</p>

## SCPI Command Reference

### *Alarms/Errors*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJect**

---

<b>Description</b>	<p>This command injects the Media Access Control (MAC) error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Mode (Manual) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ERR:MAC:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:TYPE? SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUNT

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:TYPE?**

<b>Description</b>	<p>This query gets the automated type of Media Access Control (MAC) error.</p> <p>At *RST condition, this value is set to FCS.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the automated type of MAC error.</p> <p>FCS, FCS as automated type of MAC error is selected.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE

---

<b>Description</b>	<p>This command sets the rate of Media Access Control (MAC) error to inject into the instrument.</p> <p>At *RST condition, this value is set to 1.0E-04.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Rate</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the rate of error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:ERR:MAC:AUT:RATE MIN</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUNT</pre>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the rate of Media Access Control (MAC) error to inject into the instrument.</p> <p>At *RST condition, this value is set to 1.0E-04.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Rate</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE?[ &lt;wsp&gt;MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current rate of MAC error to inject into the instrument is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Rate&gt;</code>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of MAC error to inject into the instrument.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:ERR:MAC:AUT:RATE?</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUNT?</code>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTInuous

---

<b>Description</b>	<p>This command sets the continuous Media Access Control (MAC) error to inject into the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Max Rate</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status of continuous rate/max rate.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:ERR:MAC:AUT:CONT OFF SOUR:DATA:TEL:ETH:ERR:MAC:AUT:CONT? Returns: 0</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE?</pre>

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTInuous?**

<b>Description</b>	<p>This query returns the continuous Media Access Control (MAC) error to inject into the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the continuous rate status.</p> <p>1, sets the status of continuous rate as ON.</p> <p>0, sets the status of continuous rate as OFF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:MAC:AUT:CONT ON</p> <p>SOUR:DATA:TEL:ETH:ERR:MAC:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated

---

<b>Description</b>	<p>This command set the automated error generation status.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Mode (Rate/Maxrate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status of automated error type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:MAC:AUT OFF</p> <p>SOUR:DATA:TEL:ETH:ERR:MAC:AUT?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTInuous

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated?**

<b>Description</b>	<p>This query returns the automated error generation status from the instrument.</p> <p>This query is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Mode (Rate/Maxrate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the automated error type status.</p> <p>1, returns the status of automated error type as ON.</p> <p>0, returns the status of automated error type as OFF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:MAC:AUT ON</p> <p>SOUR:DATA:TEL:ETH:ERR:MAC:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTInuous?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:HISTory?

---

<b>Description</b>	<p>This query returns the history status of the number of seconds within which physical alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Alarms</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Alarms</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:HISTory? &lt;wsp&gt;LDOWn   LFAR   LFAD   RFAult   HIBer   LOA   INVM</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LDOWn   LFAR   LFAD   RFAult   HIBer   LOA   INVM</p> <p>Selects the type of PCS alarm.</p> <p>LDOWn: LDOWn</p> <p>LFAR: LFAR</p> <p>LFAD: LFAD</p> <p>RFAult: RFAult</p> <p>HIBer: HIBer</p> <p>LOA: LOA</p> <p>INVM: InvMapping</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Reports the history status of PCS alarm. PRESENT, indicates that at least one alarm has occurred. ABSENT, indicates that no alarm occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ALAR:PHYS:GLOB:HIST? LOA
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:CURRent?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:CURRent?

---

<b>Description</b>	<p>This query returns the current status of the number of seconds within which physical alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Alarms</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Alarms</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:CURRent? &lt;wsp&gt;LDOWn   LFAR   LFAD   RFAult   HIBer   LOA   INVM   LOBL   LOAML   Skew</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LDOWn   LFAR   LFAD   RFAult   HIBer   LOA   INVM   LOBL   LOAML   Skew</p> <p>Selects the type of PCS alarm.</p> <p>LDOWn: LDOWn</p> <p>LFAR: LFAR</p> <p>LFAD: LFAD</p> <p>RFAult: RFAult</p> <p>HIBer: HIBer</p> <p>LOA: LOA</p> <p>INVM: InvMapping</p> <p>LOBL: LOBL</p> <p>LOAML: LOAML</p> <p>Skew: Skew</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Reports the current status of PCS alarm. PRESENT, indicates that at least one alarm has occurred in the last second. ABSENT, indicates that no alarm occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ALAR:PHYS:GLOB:CURR? LOA
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:HISTory?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:ETHernet:ALARm:PHYSical:GLOBal:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which physical alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Alarms</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Alarms</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:ALARm:PHYSical:GLOBal:SEConds? &lt;wsp&gt;LDOWn   LFAR   LFAD   RFAult   HIBer   LOA   INVM</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LDOWn   LFAR   LFAD   RFAult   HIBer   LOA   INVM</p> <p>Selects the type of PCS alarm.</p> <p>LDOWn: LDOWn</p> <p>LFAR: LFAR</p> <p>LFAD: LFAD</p> <p>RFAult: RFAult</p> <p>HIBer: HIBer</p> <p>LOA: LOA</p> <p>INVM: InvMapping</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the number of seconds of PCS alarm.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ALAR:PHYS:GLOB:SEC? LOA</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:ETHernet:ALARm:PHYSical:GLOBal:HISTory?</p>

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:HISTory?**

<b>Description</b>	<p>This query returns the history status for the number of seconds within which the physical alarm occurred for per lane configuration.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Alarms</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:HISTory? <wsp> <Lane>, LOBL   LOAML   Skew
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOBL   LOAML   Skew</p> <p>Selects the type of PCS alarm for each lane.</p> <p>LOBL: LOBL</p> <p>LOAML: LOAML</p> <p>Skew: Skew</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of PCS alarm for each lane.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ALAR:PHYS:HIST? 1, LOBL
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:CURREnt?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:CURRent?

---

<b>Description</b>	<p>This query returns the current status of the number of seconds within which the physical alarm occurred for PCS lanes per lane configuration.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Alarms</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:CURRent? <wsp> <Lane>, LOBL   LOAML   Skew
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOBL   LOAML   Skew</p> <p>Selects the type of PCS alarm for each lane.</p> <p>LOBL: LOBL</p> <p>LOAML: LOAML</p> <p>Skew: Skew</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of PCS alarm for each lane.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ALAR:PHYS:CURR? 1, LOBL
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:CURRent?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the physical alarm occurred for PCS lanes per lane configuration.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Alarms</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:SEConds? <wsp> <Lane>, LOBL   LOAML   Skew
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOBL   LOAML   Skew</p> <p>Selects the type of PCS alarm for each lane.</p> <p>LOBL: LOBL</p> <p>LOAML: LOAML</p> <p>Skew: Excessive Skew</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the number of seconds of PCS alarm for each lane.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ALAR:PHYS:SEC? 1, LOBL
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:SEConds?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:GLOBal:HISTory?

---

<b>Description</b>	<p>This query returns the history state within which the physical error occurred for PCS lanes.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Errors</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:GLOBal:HISTory? &lt;wsp&gt;BLOCK   INVALIDMARKER   BIP8</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   INVALIDMARKER   BIP8</p> <p>Selects the type of PCS error.</p> <p>BLOCK: BLOCK</p> <p>INVALIDMARKER: Invalid Marker</p> <p>BIP8: PCS BIP-8</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of PCS error.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:PHYS:GLOB:HIST? BIP8</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:HISTory?</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:GLOBal:CURRent?**

<b>Description</b>	<p>This query returns the current state within which the physical error occurred for particular PCS error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:GLOBal:CURRent? <wsp>BLOCK   INVALIDMARKER   BIP8
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   INVALIDMARKER   BIP8</p> <p>Selects the type of PCS error.</p> <p>BLOCK: BLOCK</p> <p>INVALIDMARKER: Invalid Marker</p> <p>BIP8: PCS BIP-8</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of PCS error.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:PHYS:GLOB:CURR? BIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:HISTory?

---

<b>Description</b>	<p>This query returns the history state within which the physical error occurred for a specified lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Errors</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Error</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:HISTory? &lt;wsp&gt; &lt;Lane&gt;, BLOCK   INVALIDMARKER   BIP8</p>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   INVALIDMARKER   BIP8</p> <p>Selects the type of PCS error for each lane.</p> <p>BLOCK: BLOCK</p> <p>INVALIDMARKER: Invalid Marker</p> <p>BIP8: PCS BIP-8</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the history status of PCS error for per lane configuration</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:PHYS:HIST? 1, BIP8</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:GLOBal:HISTory?</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:CURRent?**

<b>Description</b>	<p>This query returns the current state within which the physical error occurred for a specified lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Errors</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:CURRent? <wsp> <Lane>, BLOCK   INVALIDMARKER   BIP8
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   INVALIDMARKER   BIP8</p> <p>Selects the type of PCS error for each lane.</p> <p>BLOCK: BLOCK</p> <p>INVALIDMARKER: Invalid Marker</p> <p>BIP8: PCS BIP-8</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of PCS error for per lane configuration.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:PHYS:CURR? 1, BIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:GLOBal:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:SEConds?

<b>Description</b>	<p>This query returns the number of seconds within which the physical error occurred for a specified lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Errors</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Error</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:SEConds? &lt;wsp&gt; &lt;Lane&gt;, BLOCK   INVALIDMARKER   BIP8</p>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   INVALIDMARKER   BIP8</p> <p>Selects the type of PCS error for each lane.</p> <p>BLOCK: BLOCK</p> <p>INVALIDMARKER: Invalid Marker</p> <p>BIP8: PCS BIP-8</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the number of seconds of PCS error for per lane configuration.</p> <p>BLOCK, selects BLOCK as the type of PCS error.</p> <p>INVALIDMARKER, selects Invalid Marker as the type of PCS error.</p> <p>BIP8, selects PCS BIP-8 as the type of PCS error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:PHYS:SEC? 1, BIP8</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:GLOBal:HISTory?</p>



**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:COUNT?**

<b>Description</b>	<p>This query returns the count within which the physical error occurred for a specified lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Errors</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:COUNT? <wsp> <Lane>, BLOCK   INVALIDMARKER   BIP8
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   INVALIDMARKER   BIP8</p> <p>Selects the type of PCS error for each lane.</p> <p>BLOCK: BLOCK</p> <p>INVALIDMARKER: Invalid Marker</p> <p>BIP8: PCS BIP-8</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the count of PCS error for per lane configuration.</p> <p>BLOCK, selects BLOCK as the type of PCS error.</p> <p>INVALIDMARKER, selects Invalid Marker as the type of PCS error.</p> <p>BIP8, selects PCS BIP-8 as the type of PCS error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:PHYS:COUNT? 1, BIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:GLOBal:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:RATE?

---

<b>Description</b>	<p>This query returns the rate within which the physical error occurred for a specified lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Errors</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:RATE? <wsp> <Error>, BLOCK   INVALIDMARKER   BIP8
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   INVALIDMARKER   BIP8</p> <p>Selects the type of PCS error for each lane.</p> <p>BLOCK: BLOCK</p> <p>INVALIDMARKER: Invalid Marker</p> <p>BIP8: PCS BIP-8</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the rate of PCS error for per lane configuration.</p> <p>BLOCK, selects BLOCK as the type of PCS error.</p> <p>INVALIDMARKER, selects Invalid Marker as the type of PCS error.</p> <p>BIP8, selects PCS BIP-8 as the type of PCS error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:PHYS:RATE? 1, BIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:GLOBal:HISTory?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THReshold**

<b>Description</b>	<p>This command sets the Alarm Threshold Value for TX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Skew Alarms Threshold</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; PCS Lanes &gt; Skew Alarms Threshold</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THReshold &lt;wsp&gt; &lt;Threshold&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Threshold:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets Skew Alarm Threshold.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:ETH:ALAR:THR 20</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THReshold?**

<b>Description</b>	<p>This query returns the Alarm Threshold Value for TX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Skew Alarms Threshold</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; PCS Lanes &gt; Skew Alarms Threshold</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THReshold?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets Skew Alarm Threshold.</p> <p>This parameter is optional. If no token is specified, the current Skew Alarm Threshold value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Skew Alarm Threshold&gt;</p>
<b>Response(s)</b>	<p><b>Skew Alarm Threshold:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Skew Alarm Threshold.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ALAR:THR 20</p> <p>SOUR:DATA:TEL:ETH:ALAR:THR?</p> <p>Returns: 20</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THReshold:DEFault**

---

<b>Description</b>	<p>This command resets the Alarm Threshold Value to its default value.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Skew Alarms Threshold Default (button)</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; PCS Lanes &gt; Skew Alarms Threshold Default (button)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THReshold:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ALAR:THR:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THReshold

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE**

<b>Description</b>	<p>This command sets the type of physical alarm for the instrument.</p> <p>At *RST condition, this value is set to LDOWn.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE &lt;wsp&gt;LDOWn   LFAult   RFAult</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LDOWn   LFAult   RFAult</p> <p>Sets the type of physical alarm for the instrument.</p> <p>LDOWn: Link Down</p> <p>LFAult: Local Fault</p> <p>RFAult: Remote Fault</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE LFA</p> <p>SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE?</p> <p>Returns: LFAULT</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE?**

<b>Description</b>	<p>This query returns the type of physical alarm for the instrument.</p> <p>At *RST condition, this value is set to LDOWn.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of physical alarm.</p> <p>LDOWN, Link Down as the alarm type is selected.</p> <p>LFAULT, Local Fault as the alarm type is selected.</p> <p>RFAULT, Remote Fault as the alarm type is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE LFA</p> <p>SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE?</p> <p>Returns: LFAULT</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE**

<b>Description</b>	<p>This command sets the manual type physical error.</p> <p>At *RST condition, this value is set to INVALIDMARKER.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE &lt;wsp&gt;BLOCk   INVALIDMARKER   BIP8</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCk   INVALIDMARKER   BIP8</p> <p>Sets the manual type physical error.</p> <p>BLOCk: Block</p> <p>INVALIDMARKER: Invalid Marker</p> <p>BIP8: PCS BIP8</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE BLOC</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE?</p> <p>Returns: BLOCk</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:INJECT</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE?**

---

<b>Description</b>	<p>This query returns the manual type physical error.</p> <p>At *RST condition, this value is set to BLOCk.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of physical error.</p> <p>BLOCK, Block as the type of physical error is selected.</p> <p>INVALIDMARKER, Invalid Marker as the type of physical error is selected.</p> <p>BIP8, PCS BIP8 as the type of physical error is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE BLOC</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE?</p> <p>Returns: BLOCk</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:INJECT</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT

---

<b>Description</b>	<p>This command sets the amount of physical error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of physical error to inject.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:AMO 50</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:AMO?</p> <p>Returns: 50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:INJECT</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUt?**

<b>Description</b>	<p>This query returns the amount of physical error injected.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUt?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the amount of physical error injected.</p> <p>This parameter is optional. If no token is specified, the current amount of physical error injected value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of physical error injected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:AMO 50</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:AMO?</p> <p>Returns: 50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUt</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:INJect</p>

## SCPI Command Reference

### *Alarms/Errors*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:INJect**

---

<b>Description</b>	<p>This command injects the physical error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ERR:PHYS:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:TYPE? SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUnt?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:TYPE**

---

<b>Description</b>	<p>This command sets the automated type of pattern error.</p> <p>At *RST condition, this value is set to Invalid Marker.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Defect</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:TYPE &lt;wsp&gt;BLOCk   INVALIDMARKER   BIP8</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCk   INVALIDMARKER   BIP8</p> <p>Selects the AUTOMATED type pattern error.</p> <p>BLOCk: Block</p> <p>INVALIDMARKER: Invalid Marker</p> <p>BIP8: PCS BIP-8</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE BLOCk</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the automated type of pattern error.</p> <p>At *RST condition, this value is set to Invalid Marker.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the automated type pattern error.</p> <p>BLOCK, Block as the type of physical error is selected.</p> <p>INVALIDMARKER, Invalid Marker as the type of physical error is selected.</p> <p>BIP8, PCS BIP8 as the type of physical error is selected.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:RATE**

<b>Description</b>	<p>This command sets the rate of pattern error to inject into the instrument.</p> <p>At *RST condition, this value is set to 1.0E-02.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Rate</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the rate of pattern error to inject into the instrument.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:RATE MIN</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMount</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the rate of pattern error to inject into the instrument.</p> <p>At *RST condition, this value is set to 1.0E-02.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Bert</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:RATE? [ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Returns the rate of pattern error to inject into the instrument.</p> <p>This parameter is optional. If no token is specified, the current rate of pattern error to inject into the instrument is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Error&gt;</p>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of pattern error to inject into the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:RATE?</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUnt?</p>



**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTInuous**

<b>Description</b>	<p>This command sets the continuous pattern error to inject into the instrument.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Max Rate</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the continuous pattern error.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:CONT OFF SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:CONT? Returns: 0</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTInuous?

---

<b>Description</b>	<p>This query returns the continuous pattern error to inject into the instrument.</p> <p>This query is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the continuous pattern error to inject into the instrument.</p> <p>1, sets the continuous pattern error as enabled.</p> <p>0, sets the continuous pattern error as disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:CONT ON</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated?

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated**

<b>Description</b>	<p>This command sets the automated pattern type.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the automated pattern type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:AUT OFF</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:AUT?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTInuous

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated?

---

<b>Description</b>	<p>This query returns the automated pattern type.</p> <p>This query is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the automated pattern type.</p> <p>1, returns the automated pattern type as ON.</p> <p>0, returns the automated pattern type as OFF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:AUT ON</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTinuous?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:LINK?**

<b>Description</b>	<p>This query returns the status of Ethernet alarms.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; 100GE/40GE &gt; Interface</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:LINK? <wsp>LDOWn   LFAR   LFAD   RFAult   HIBer   LOA   INVM
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LDOWn   LFAR   LFAD   RFAult   HIBer   LOA   INVM</p> <p>Selects the alarm type whose status is to be retrieved.</p> <p>LDOWn: LDOWn.</p> <p>LFAR: LFAR.</p> <p>LFAD: LFAD.</p> <p>RFAult: RFAult.</p> <p>HIBer: HIBer.</p> <p>LOA: LOA.</p> <p>INVM: InvMapping.</p>
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the status of the Ethernet alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ALAR:LINK? LFAR
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATTern:ALARm:SYNCh?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical

---

<b>Description</b>	<p>This command enables or disables the status of the Ethernet alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Ethernet &gt; Alarms</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Ethernet alarm generation.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ALAR:PHYS ON</p> <p>SOUR:DATA:TEL:ETH:ALAR:PHYS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:AUTomated?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical?**

---

<b>Description</b>	<p>This query returns the status of Ethernet alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Ethernet &gt; Alarms</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Ethernet alarm generation.</p> <p>1, alarm generation enabled.</p> <p>0, alarm generation disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ALAR:PHYS OFF</p> <p>SOUR:DATA:TEL:ETH:ALAR:PHYS?</p> <p>Returns: 0</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:AUTomated?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:COUNT:TOTal?

---

<b>Description</b>	<p>This query returns the total count within which physical error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Errors &gt; PCS Lanes</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:COUNT:TOTal? <wsp>BLOCK   INVALIDMARKER   BIP8
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   INVALIDMARKER   BIP8</p> <p>Selects the type of PCS error.</p> <p>BLOCK: BLOCK</p> <p>INVALIDMARKER: INVALIDMARKER</p> <p>BIP8: BIP8</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Total count of PCS error for per lane configuration.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:PHYS:COUN:TOT? BIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:GLOBal:HISTory?

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:RATE:TOTal?**

<b>Description</b>	<p>This query returns the rate within which physical error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; PCS Lanes &gt; Errors &gt; PCS Lanes</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:RATE:TOTal? <wsp>BLOCK   INVALIDMARKER   BIP8
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   INVALIDMARKER   BIP8</p> <p>Selects the type of PCS error.</p> <p>BLOCK: BLOCK</p> <p>INVALIDMARKER: INVALIDMARKER</p> <p>BIP8: BIP8</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Total rate of PCS error for per lane configuration.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:PHYS:RATE:TOT? BIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:GLOBal:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:LANE

---

<b>Description</b>	<p>This command sets the particular lane number for the injection purpose for PCS errors.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Ether Bert) &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Lane</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:LANE <wsp> <Lane>, ON   OFF
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which injection is to be done for PCS errors.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the per lane status for PCS errors.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:LANE 1, ON</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:LANE? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:ALANes?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:LANE?**

---

<b>Description</b>	<p>This query returns the particular lane number for the injection purpose for PCS errors.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Ether Bert) &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Lane</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:LANE? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which injection is to be done for PCS errors.</p>
<b>Response Syntax</b>	<status>
<b>Response(s)</b>	<p><b>status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of particular lane for PCS errors.</p> <p>1, returns the status of a particular lane as enabled.</p> <p>0, returns the status of a particular lane as disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:LANE 1, ON</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:LANE? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:ALANes

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:ALANes**

<b>Description</b>	<p>This command sets all lanes for the injection purpose for PCS errors.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Ether Bert) &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; All Lanes</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:ALANes <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the all lane status for PCS errors.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:ALAN ON</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:ALAN?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:LANE?

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:ALANes?**

<b>Description</b>	<p>This query returns all lanes for the injection purpose for PCS errors.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test (Ether Bert) &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (PCS) &gt; Lane</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:ALANes?
<b>Response Syntax</b>	<status>
<b>Response(s)</b>	<p><b>status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of PCS errors for all lanes.</p> <p>1, returns the status of all lanes as enabled.</p> <p>0, returns the status of all lanes as disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:PHYS:ALAN ON</p> <p>SOUR:DATA:TEL:ETH:ERR:PHYS:ALAN?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:LANE

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:ETHernet:ALARm:LRATe:GLOBal:CURRent?

---

<b>Description</b>	<p>This query returns the current status of the ethernet alarm occurred for lower rates.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Alarms</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical)</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELecom:ETHernet:ALARm:LRATe:GLOBal:CURRent? &lt;wsp&gt;LDOWN   LFAULTRECIEVED   LFAULTDETECTED   RFAULT</pre>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LDOWN   LFAULTRECIEVED   LFAULTDETECTED   RFAULT</p> <p>Selects the type of Ethernet alarm.</p> <p>LDOWN: LDOWN</p> <p>LFAULTRECIEVED: LFAULTRECIEVED</p> <p>LFAULTDETECTED: LFAULTDETECTED</p> <p>RFAULT: RFAULT</p>
<b>Response Syntax</b>	<Current Status>
<b>Response(s)</b>	<p><b>Current Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of ethernet alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ALAR:LRAT:GLOB:CURR? LDOWN
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:ALARm:PHYSical:GLOBal:HISTory?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:GLOBal:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which ethernet alarm occurred for lower rates.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Alarms</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:GLOBal:SEConds? <wsp>LDOWn   LFAR   LFAD   RFAult
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LDOWn   LFAR   LFAD   RFAult</p> <p>Selects the type of Ethernet alarm.</p> <p>LDOWn: LDOWn</p> <p>LFAR: LFAR</p> <p>LFAD: LFAD</p> <p>RFAult: RFAult</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the number of seconds of ethernet alarm.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ALAR:LRAT:GLOB:SEC? LDOWn
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:LRATe:GLOBal:HISTory?

---

<b>Description</b>	<p>This query returns the history status of the ethernet error occurred (Lower rates). At *RST condition, this value is set to device-dependent. Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Errors Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:LRATe:GLOBal:HISTory?<wsp>BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL
<b>Parameter(s)</b>	<p><b>Error:</b> The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element. The allowed elements for this parameter are: BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL Selects the type of Ethernet error. BLOCK: BLOCK TERROR: Total Error FCARRIER: False carrier IDLE1: Idle SYMBOL: Symbol</p>
<b>Response Syntax</b>	<History Status>
<b>Response(s)</b>	<p><b>History Status:</b> The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element. Reports the history status of ethernet error. PRESENT, indicates that at least one error has occurred. ABSENT, indicates that no error occurred. INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:LRAT:GLOB:HIST? BLOCK
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:HISTory?

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:CURRent?**

<b>Description</b>	<p>This query returns the current status of the ethernet error (Lower rates).</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Errors</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical)</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:CURRent? &lt;wsp&gt;BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL</p> <p>Selects the type of Ethernet error.</p> <p>BLOCK: BLOCK</p> <p>TERROR: Total Error</p> <p>FCARRIER: False carrier</p> <p>IDLE1: Idle</p> <p>SYMBOL: Symbol</p>
<b>Response Syntax</b>	<Current Status>
<b>Response(s)</b>	<p><b>Current Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of ethernet error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:LRAT:GLOB:CURR? BLOCK
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:LRATe:GLOBal:SEConds?

---

<b>Description</b>	<p>This query returns the seconds within which the ethernet error occurred (Lower rates). At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Alarms Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:LRATe:GLOBal:SEConds? &lt;wsp&gt;BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL</p> <p>Selects the type of Ethernet error.</p> <p>BLOCK: BLOCK TERROR: Total Error FCARRIER: False carrier IDLE1: Idle SYMBOL: Symbol</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the Seconds of ethernet error.</p> <p>PRESENT, indicates that at least one error has occurred. ABSENT, indicates that no error occurred. INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:ERR:LRAT:GLOB:SEC? BLOCK</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:ETHernet:ERRor:PHYSical:HISTory?</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:COUNT?**

<b>Description</b>	<p>This query returns the Count within of the ethernet error occurred (Lower rates). At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Alarms Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:COUNT? <wsp>BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL</p> <p>Selects the type of Ethernet error.</p> <p>BLOCK: BLOCK</p> <p>TERROR: Total Error</p> <p>FCARRIER: False carrier</p> <p>IDLE1: Idle</p> <p>SYMBOL: Symbol</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the count of ethernet error.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:LRAT:GLOB:COUN? BLOCK
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:RATE?

---

<b>Description</b>	<p>This query returns the rate of ethernet error occurred (Lower rates). At *RST condition, this value is set to device-dependent. Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Errors Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:GLOBal:RATE? <wsp>BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL
<b>Parameter(s)</b>	<p><b>Error:</b> The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element. The allowed elements for this parameter are: BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL Selects the type of Ethernet error. BLOCK: BLOCK TERROR: Total Error FCARRIER: False carrier IDLE1: Idle SYMBOL: Symbol</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b> The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element. Reports the rate of ethernet error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:ERR:LRAT:GLOB:RATE? BLOCK
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:GLOBal:HISTory? LOA

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:MANual:TYPE**

<b>Description</b>	<p>This query returns the manual type Ethernet error.</p> <p>At *RST condition, this value is set to FCS.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Defect</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:MANual:TYPE <wsp>BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: BLOCK   TERROR   FCARRIER   IDLE1   SYMBOL</p> <p>Selects the type of Ethernet error.</p> <p>BLOCK: BLOCK</p> <p>TERROR: Total Error</p> <p>FCARRIER: False carrier</p> <p>IDLE1: Idle</p> <p>SYMBOL: Symbol</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ERR:LRATe:MAN:TYPE SYMBOL
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJECT</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type Ethernet error.</p> <p>At *RST condition, this value is set to FCS.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Defect</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the manual type Ethernet error.</p> <p>SYMBOL, SYMBOL as manual type Ethernet error is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:LRATE:MAN:TYPE?</p> <p>Returns: SYMBOL</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUnt</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJect</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AMOUnt**

<b>Description</b>	<p>This command sets the amount of Ethernet error to inject into the instrument.</p> <p>At *RST condition, this value is set to MINimum.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Amount</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AMOUnt &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:ERR:LRATe:AMO 50 SOUR:DATA:TEL:ETH:ERR:LRATe:AMO? Returns: 50</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUnt? SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJect</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AMOut?

---

<b>Description</b>	<p>This query returns the amount of Ethernet error to inject.</p> <p>At *RST condition, this value is set to MINimum.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Amount</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AMOut?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of Ethernet error to inject into the instrument is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of Ethernet error to inject into the instrument.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:LRATe:AMO 50</p> <p>SOUR:DATA:TEL:ETH:ERR:LRATe:AMO?</p> <p>Returns: 50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOut</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJect</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:INJect**

---

<b>Description</b>	<p>This command injects the Ethernet error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Inject</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ERR:LRATe:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:TYPE? SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUNT

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:TYPE**

<b>Description</b>	<p>This command gets the automated type of Ethernet error.</p> <p>At *RST condition, this value is set to FCS.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Defect</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:TYPE &lt;wsp&gt;SYMBOL</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SYMBOL</p> <p>Selects the type of Ethernet error.</p> <p>SYMBOL</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:LRATe:AUT:TYPE SYMBOL</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:TYPE?**

<b>Description</b>	<p>This command gets the automated type of Ethernet error.</p> <p>At *RST condition, this value is set to FCS.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Defect</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the automated type Ethernet error.</p> <p>SYMBOL, SYMBOL as automated type Ethernet error is selected.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ERR:LRATe:AUT:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:RATE

---

<b>Description</b>	<p>This command sets the rate of Ethernet error to inject into the instrument.</p> <p>At *RST condition, this value is set to 1.0E-04.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Rate Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the rate of error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:ETH:ERR:LRATE:AUT:RATE MIN</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUNT</pre>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the rate of Ethernet error to inject into the instrument.</p> <p>At *RST condition, this value is set to 1.0E-04.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Rate Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:RATE?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current rate of Ethernet error to inject into the instrument is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of Ethernet error to inject into the instrument.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ERR:LRATE:AUT:RATE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUnt?

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:CONTInuous**

<b>Description</b>	<p>This command sets the continuous Ethernet error to inject into the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Max Rate</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status of continuous rate/max rate.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:LRATe:AUT:CONT OFF</p> <p>SOUR:DATA:TEL:ETH:ERR:LRATe:AUT:CONT ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE?</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:CONTInuous?**

<b>Description</b>	<p>This query returns the continuous Ethernet error to inject into the instrument.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Max Rate</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the continuous rate status.</p> <p>1, Continuous rate is enabled.</p> <p>0, Continuous rate is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:ERR:LRATe:AUT:CONT?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated

---

<b>Description</b>	<p>This command set the automated Ethernet error generation status in the instrument.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Inject</p> <p>Rates (10/100/100M Electrical, 10GE LAN, 10GE WAN, 100 M Optical, 1 GE Optical, OTU3)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:LRATe:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status of automated error type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:ERR:LRATe:AUT ON</p> <p>SOUR:DATA:TEL:ETH:ERR:LRATe:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTinuous

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe**

---

<b>Description</b>	<p>This command enable/disable the alarm generation status.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt;Global Injection &gt; Layer (Ethernet) &gt;Inject</p> <p>Rates (10GE LAN, 10GE WAN, OTU3)</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the alarm generation status.</p>
<b>Example(s)</b>	<pre>SOURce:DATA:TELEcom:ETHernet:ALARm:LRATe:TYPE LDOWN SOURce:DATA:TELEcom:ETHernet:ALARm:LRATe ON SOURce:DATA:TELEcom:ETHernet:ALARm:LRATe? Returns: 1</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:TYPE</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe?**

<b>Description</b>	<p>This query returns the alarm generation status.</p> <p>This query is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Inject</p> <p>Rates (10GE LAN, 10GE WAN, OTU3)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the alarm generation status.</p> <p>1, Alarm generation is enabled.</p> <p>0, Alarm generation is disabled.</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ALARm:LRATe:TYPE LDOWN</p> <p>SOURce:DATA:TELEcom:ETHernet:ALARm:LRATe ON</p> <p>SOURce:DATA:TELEcom:ETHernet:ALARm:LRATe?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:TYPE?

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:TYPE**

<b>Description</b>	<p>This command set the Ethernet Alarm type.</p> <p>*RST condition, Value is set to LDOWN.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (Ethernet) &gt; Defect</p> <p>Rates (10GE LAN, 10GE WAN, OTU3)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe:TYPE <wsp>LDOWN   LFAULT   RFAULT   LFAULTRECIEVED   LFAULTDETECTED
<b>Parameter(s)</b>	<p><b>Alarm Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LDOWN   LFAULT   RFAULT   LFAULTRECIEVED   LFAULTDETECTED</p> <p>Sets the alarm type.</p> <p>LDOWN: LDOWN</p> <p>LFAULT: LFAULT</p> <p>RFAULT: RFAULT</p> <p>LFAULTRECIEVED: LFAULTRECIEVED</p> <p>LFAULTDETECTED: LFAULTDETECTED</p>
<b>Example(s)</b>	<p>SOURce:DATA:TELEcom:ETHernet:ALARm:LRATe:TYPE LDOWN</p> <p>SOURce:DATA:TELEcom:ETHernet:ALARm:LRATe:TYPE?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:LRATe

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:OTN:ERRor:PHYSical:GLOBal:HISTory?

---

<b>Description</b>	<p>This query returns the history status for the Alarm Generation for Global OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ERRor:PHYSical:GLOBal:HISTory? <wsp>FAS   LLM
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the error for the history status.</p> <p>FAS</p> <p>LLM</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of the physical error.</p> <p>PRESENT, indicates that at least one physical error is present.</p> <p>ABSENT, indicates that there is no physical error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ERR:PHYS:GLOB:HIST? FAS</p> <p>FETC:DATA:TEL:OTN:ERR:PHYS:GLOB:HIST? LLM</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ERRor:PHYSical:GLOBal:CURRent?

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:GLOBal:CURRent?**

<b>Description</b>	<p>This query returns Current Status for Global OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:GLOBal:CURRent? <wsp>FAS   LLM
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the error for the current status.</p> <p>FAS</p> <p>LLM</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of physical error.</p> <p>PRESENT, indicates that at least one physical error is present.</p> <p>ABSENT, indicates that there is no physical error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ERR:PHYS:GLOB:CURR? FAS</p> <p>FETC:DATA:TEL:OTN:ERR:PHYS:GLOB:CURR? LLM</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:GLOBal:HIStory?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:HISTory?

---

<b>Description</b>	<p>This query returns the history status physical error for lane OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:HISTory? &lt;wsp&gt; &lt;Lane&gt;, FAS   LLM</p>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the history status of physical error. The range for the virtual lane is from 0 to 19.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the error for the error history.</p> <p>FAS: FAS</p> <p>LLM: LLM</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of physical error.</p> <p>PRESENT, indicates that at least one physical error is present.</p> <p>ABSENT, indicates that there is no physical error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ERR:PHYS:HIST? 1, FAS</p> <p>FETC:DATA:TEL:OTN:ERR:PHYS:HIST? 1, LLM</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:CURREnt?</p>

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:CURRent?**

<b>Description</b>	<p>This query returns the current status physical error for lane OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:CURRent? <wsp> <Lane>, FAS   LLM
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the current status of physical error. The range for the virtual lane is from 0 to 19.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the error for the current.</p> <p>FAS: FAS</p> <p>LLM: LLM</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of physical error.</p> <p>PRESENT, indicates that at least one physical error is present.</p> <p>ABSENT, indicates that there is no physical error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ERR:PHYS:CURR? 1, FAS</p> <p>FETC:DATA:TEL:OTN:ERR:PHYS:CURR? 1, LLM</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:HIStory?

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELecom:OTN:ERRor:PHYSical:SEConds?

<b>Description</b>	<p>This query returns the number of seconds the physical error was raised for lane OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ERRor:PHYSical:SEConds? <wsp> <Lane>, FAS   LLM
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the number of seconds of physical error. The range for the virtual lane is from 0 to 19.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the error for the error seconds.</p> <p>FAS: FAS</p> <p>LLM: LLM</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of physical error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ERR:PHYS:SEC? 1, FAS</p> <p>FETC:DATA:TEL:OTN:ERR:PHYS:SEC? 1, LLM</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ERRor:PHYSical:RATE?



---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:COUNT?**

---

<b>Description</b>	<p>This query returns the count of physical error for the Lane OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:COUNT? &lt;wsp&gt; &lt;Lane&gt;, FAS   LLM</code>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the count of physical error. The range for the virtual lane is from 0 to 19.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the error for the error count.</p> <p>FAS: FAS</p> <p>LLM: LLM</p>
<b>Response Syntax</b>	<code>&lt;Count&gt;</code>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of physical error.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:OTN:ERR:PHYS:COUN? 1, FAS</code> <code>FETC:DATA:TEL:OTN:ERR:PHYS:COUN? 1, LLM</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:RATE?</code>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:RATE?

---

<b>Description</b>	<p>This query returns the current rate of physical error for Lane OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:RATE? <wsp> <Lane>, FAS   LLM
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the current rate of physical error. The range for the virtual lane is from 0 to 19.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the error for the error rate.</p> <p>FAS</p> <p>LLM</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of physical error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ERR:PHYS:RATE? 1, FAS</p> <p>FETC:DATA:TEL:OTN:ERR:PHYS:RATE? 1, LLM</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:COUNT?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:COUNT:TOTal?**

---

<b>Description</b>	<p>This query returns the total count of the OTL lane errors.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: OTN BERT Test &gt; Results &gt; Alarms/Errors &gt; OTL &gt; Errors (FAS, INVALID MARKER)</p> <p>Navigation Path: OTN BERT Test &gt; Functions &gt;40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Error (FAS, INVALID MARKER)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:COUNT:TOTal? <wsp>FAS   LLM
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the type of OTL error.</p> <p>FAS</p> <p>LLM</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the Total count of OTL error for per lane configuration.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:PHYS:COUN:TOT? FAS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:COUNT?

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:RATE:TOTal?**

<b>Description</b>	<p>This query returns the total rate of the OTL lane errors.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: OTN BERT Test &gt; Results &gt; Alarms/Errors &gt; OTL &gt; Errors (FAS, INVALID MARKER)</p> <p>Navigation Path: OTN BERT Test &gt; Functions &gt;40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Error (FAS, INVALID MARKER)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:RATE:TOTal? <wsp>FAS   LLM
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the type of OTL error.</p> <p>FAS</p> <p>LLM</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the total rate of OTL error for per lane configuration.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:PHYS:RATE:TOT? FAS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:RATE?

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:GLOBal:HISTory?**

<b>Description</b>	<p>This query returns the history status for the Alarm Generation for Global OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:GLOBal:HISTory? <wsp>LOL   LOF   OOF   LOR   OOR
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOL   LOF   OOF   LOR   OOR</p> <p>Selects the alarm for the history status.</p> <p>LOL: the LOL for the history status of Global Physical OTL Alarm.</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of physical alarm.</p> <p>PRESENT, indicates that at least one physical error is present.</p> <p>ABSENT, indicates that there is no physical error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:PHYS:GLOB:HIST? LOL
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:GLOBal:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:GLOBal:CURRent?

---

<b>Description</b>	<p>This query returns the Current Status for the Alarm Generation for Global OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:GLOBal:CURRent? &lt;wsp&gt;LOL   LOF   OOF   LOR   OOR</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOL   LOF   OOF   LOR   OOR</p> <p>Selects the alarm for the current status.</p> <p>LOL: LOL for the Current status of Global Physical OTL Alarm.</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of physical alarm.</p> <p>PRESENT, indicates that at least one physical error is present.</p> <p>ABSENT, indicates that there is no physical error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ALAR:PHYS:GLOB:CURR? LOL</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:GLOBal:HISTory?</p>

---

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:PHYSical:GLOBal:SEConds?**

---

<b>Description</b>	<p>This query returns seconds for Global OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL &gt; LOL</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:OTN:ALARm:PHYSical:GLOBal:SEConds? &lt;wsp&gt;LOL   LOF   OOF   LOR   OOR</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOL   LOF   OOF   LOR   OOR</p> <p>Selects the alarm for the seconds value retrieval.</p> <p>LOL: LOL for the seconds value of physical alarm.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the seconds value of physical alarm.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ALAR:PHYS:GLOB:SEC? LOL</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:ERRor:PHYSical:SEConds?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:HISTory?

---

<b>Description</b>	<p>This query returns the history status of physical alarm for lane OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:HISTory? &lt;wsp&gt; &lt;Lane&gt;, OOF   LOF   OOR   LOR   SKEW</p>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the history status of physical alarm. The range for the virtual lane is from 0 to 19.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OOF   LOF   OOR   LOR   SKEW</p> <p>Selects the alarm for the lane alarm history.</p> <p>OOF: OOF for the history status of physical alarm.</p> <p>LOF: LOF for the history status of physical alarm.</p> <p>OOR: OOR for the history status of physical alarm.</p> <p>LOR: AIS for the history status of physical alarm.</p> <p>SKEW: SKEW for the history status of physical alarm.</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>

---



**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of physical lane alarm. PRESENT, indicates that at least one physical error is present. ABSENT, indicates that there is no physical error. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:PHYS:HIST? 1, SKEW FETC:DATA:TEL:OTN:ALAR:PHYS:HIST? 1, LOF
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:CURRent?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:OTN:ALARm:PHYSical:CURRent?

---

<b>Description</b>	<p>This query returns the current status of physical alarm for lane OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELecom:OTN:ALARm:PHYSical:CURRent? &lt;wsp&gt; &lt;Lane&gt;, OOF   LOF   OOR   LOR   SKEW</code>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the current status of physical alarm. The range for the virtual lane is from 0 to 19.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OOF   LOF   OOR   LOR   SKEW</p> <p>Selects the alarm for the per lane current alarm.</p> <p>OOF: OOF for the Current status of physical alarm.</p> <p>LOF: LOF for the Current status of physical alarm.</p> <p>OOR: OOR for the Current status of physical alarm.</p> <p>LOR: AIS for the Current status of physical alarm.</p> <p>SKEW: SKEW for Current status of physical alarm.</p>
<b>Response Syntax</b>	<code>&lt;Current&gt;</code>

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:PHYSical:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of physical lane alarm. PRESENT, indicates that at least one physical error is present. ABSENT, indicates that there is no physical error. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:PHYS:CURR? 1, SKEW FETC:DATA:TEL:OTN:ALAR:PHYS:CURR? 1, OOF
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:PHYSical:HIStory?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:OTN:ALARm:PHYSical:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds the physical alarm was raised for lane OTL Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTL</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:OTN:ALARm:PHYSical:SEConds? &lt;wsp&gt; &lt;Lane&gt;, OOF   LOF   OOR   LOR   SKEW</p>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the number of seconds of physical alarm. The range for the virtual lane is from 0 to 19.</p> <p><b>Alarm:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OOF   LOF   OOR   LOR   SKEW</p> <p>Selects the alarm for the per lane seconds value retrieval.</p> <p>OOF: OOF for the seconds value of physical alarm.</p> <p>LOF: LOF for the seconds value of physical alarm.</p> <p>OOR: OOR for the seconds value of physical alarm.</p> <p>LOR: AIS for the seconds value of physical alarm.</p> <p>SKEW: SKEW for the seconds value of physical alarm.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of physical lane alarm.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ALAR:PHYS:SEC? 1, SKEW</p> <p>FETC:DATA:TEL:OTN:ALAR:PHYS:SEC? 1, OOR</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:ERRor:PHYSical:RATE?</p>

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:HISTory?**

<b>Description</b>	<p>This query returns the history status of the Optical Transport Unit (OTU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:HISTory? <wsp>LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIAe
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIAe</p> <p>Selects the type of OTU (Optical Transport Unit) alarm.</p> <p>LOF: LOF (Loss Of Frame) when OOF is present for at least 3 ms.</p> <p>OOF: OOF (Out-Of-Frame) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU (Optical Transport Unit) frames.</p> <p>LOM: LOM (Loss Of Multiframe) when OOM (Out of Multiframe) is present for at least 3 ms.</p> <p>OOM: OOM (Out-Of-Multiframe) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.</p> <p>OAIS: OTU-AIS (OTU - Alarm Indication Signal) when polynomial number 11 (PN-11) is over all OTU (Optical Transport Unit) frame bits including FAS and MFAS (Multiframe Alignment Signal) for at least 3 consecutive 8192 bit-interval.</p> <p>OTIM: OTU-TIM (OTU - Trace Identifier Mismatch) when expected SM SAPI (Source Access Point Identifier) and/or SM DAPI (Destination Access Point Identifier) do not match the received SM SAPI and/or DAPI for at least 3 consecutive TTI (Trail Trace Identifier) of the 256 frames multiframe.</p> <p>OBDI: OTU-BDI (OTU - Backward Defect Indication) when the BDI (Backward Defect Indication) bit in the SM overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OIAE: OTU-IAE (OTU - Incoming Alignment Error) when IAE bit in the SM overhead field (byte 3, bit 6) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OBIAe: OTU-BIAE (OTU - Backward Incoming Alignment Error) when BEI (Backward Error Indication) /BIAE (Backward Incoming Alignment Error) bits in the SM overhead field (byte 3, bits 1 to 4) are ""1011"" for at least 3 consecutive frames.</p>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:HISTory?**

**Response Syntax** <History>

**Response(s)** **History:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the history status of the Optical Transport Unit (OTU) alarm.

PRESENT, indicates that at least one alarm has occurred.

ABSENT, indicates that no alarm occurred.

INACTIVE, indicates that the test did not run yet.

**Example(s)** FETC:DATA:TEL:OTN:ALAR:OTU1:HIST? OAIS

**See Also** FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:SEConds?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:SEConds?**

---

<b>Description</b>	<p>This query returns the number of seconds within which the Optical Transport Unit (OTU) alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:SEConds? &lt;wsp&gt;LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIAe</p>

---

## SCPI Command Reference

### Alarms/Errors

---

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:SEConds?

Parameter(s)	Alarm:
	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIAe</p> <p>Selects the type of OTU (Optical Transport Unit) alarm.</p> <p>LOF: LOF (Loss Of Frame) when OOF is present for at least 3 ms.</p> <p>OOF: OOF (Out-Of-Frame) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU (Optical Transport Unit) frames.</p> <p>LOM: LOM (Loss Of Multiframe) when OOM (Out of Multiframe) is present for at least 3 ms.</p> <p>OOM: OOM (Out-Of-Multiframe) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.</p> <p>OAIS: OTU-AIS (OTU - Alarm Indication Signal) when polynomial number 11 (PN-11) is over all OTU (Optical Transport Unit) frame bits including FAS and MFAS (Multiframe Alignment Signal) for at least 3 consecutive 8192 bit-interval.</p> <p>OTIM: OTU-TIM (OTU - Trace Identifier Mismatch) when expected SM SAPI (Source Access Point Identifier) and/or SM DAPI (Destination Access Point Identifier) do not match the received SM SAPI and/or DAPI for at least 3 consecutive TTI (Trail Trace Identifier) of the 256 frames multiframe.</p> <p>OBDI: OTU-BDI (OTU - Backward Defect Indication) when the BDI (Backward Defect Indication) bit in the SM overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OIAE: OTU-IAE (OTU - Incoming Alignment Error) when IAE bit in the SM overhead field (byte 3, bit 6) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OBIAe: OTU-BIAE (OTU - Backward Incoming Alignment Error) when BEI (Backward Error Indication) /BIAE (Backward Incoming Alignment Error) bits in the SM overhead field (byte 3, bits 1 to 4) are ""1011"" for at least 3 consecutive frames.</p>

**Response Syntax** <Seconds>

---



---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:SEConds?**

---

<b>Response(s)</b>	<b>Seconds:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Transport Unit (OTU) alarm.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:OTU1:SEC? OAI5
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:HISTory?

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:CURRent?**

<b>Description</b>	<p>This query returns the current status of the Optical Transport Unit (OTU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:CURRent? &lt;wsp&gt;LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIAe</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIAe</p> <p>Selects the type of OTU (Optical Transport Unit) alarm.</p> <p>LOF: LOF (Loss Of Frame) when OOF is present for at least 3 ms.</p> <p>OOF: OOF (Out-Of-Frame) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU (Optical Transport Unit) frames.</p> <p>LOM: LOM (Loss Of Multiframe) when OOM (Out of Multiframe) is present for at least 3 ms.</p> <p>OOM: OOM (Out-Of-Multiframe) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.</p> <p>OAIS: OTU-AIS (OTU - Alarm Indication Signal) when polynomial number 11 (PN-11) is over all OTU (Optical Transport Unit) frame bits including FAS and MFAS (Multiframe Alignment Signal) for at least 3 consecutive 8192 bit-interval.</p> <p>OTIM: OTU-TIM (OTU - Trace Identifier Mismatch) when expected SM SAPI (Source Access Point Identifier) and/or SM DAPI (Destination Access Point Identifier) do not match the received SM SAPI and/or DAPI for at least 3 consecutive TTI (Trail Trace Identifier) of the 256 frames multiframe.</p> <p>OBDI: OTU-BDI (OTU - Backward Defect Indication) when the BDI (Backward Defect Indication) bit in the SM overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OIAE: OTU-IAE (OTU - Incoming Alignment Error) when IAE bit in the SM overhead field (byte 3, bit 6) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OBIAe: OTU-BIAE (OTU - Backward Incoming Alignment Error) when BEI (Backward Error Indication) /BIAE (Backward Incoming Alignment Error) bits in the SM overhead field (byte 3, bits 1 to 4) are ""1011"" for at least 3 consecutive frames.</p>

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:CURRent?**

---

**Response Syntax** <Current>**Response(s)** **Current:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current status of the Optical Transport Unit (OTU) alarm.

PRESENT, indicates that at least one alarm has occurred in the last second.

ABSENT, indicates that there is no alarm.

INACTIVE, indicates that the test did not run yet.

**Example(s)** FETC:DATA:TEL:OTN:ALAR:OTU1:CURR? OAIS**See Also** FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:HISTory?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### **:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:HISTory?**

---

<b>Description</b>	<p>This query returns the history status of Optical Transport Unit (OTU) alarm for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU3e(1/2)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:HISTory? &lt;wsp&gt;LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIAe</p>

---

**:FETCH[1..n]:DATA:TELEcom:OTN:ALARM:OTU[1..n]:E[1..n]:HISTORY?**

<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIAe</p> <p>Selects the Optical Transport Unit (OTU) alarm type.</p> <p>LOF: Loss Of Frame (LOF) when OOF is present for at least 3 ms.</p> <p>OOF: Out-Of-Frame (OOF) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU (Optical Transport Unit) frames.</p> <p>LOM: Loss Of Multiframe (LOM) when OOM (Out of Multiframe) is present for at least 3 ms.</p> <p>OOM: Out-Of-Multiframe (OOM) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.</p> <p>OAIS: OTU - Alarm Indication Signal (OTU-AIS) when polynomial number 11 (PN-11) is over all OTU (Optical Transport Unit) frame bits including FAS and MFAS (Multiframe Alignment Signal) for at least 3 consecutive 8192 bit-interval.</p> <p>OTIM: OTU - Trace Identifier Mismatch (OTU-TIM) when expected SM SAPI (Source Access Point Identifier) and/or SM DAPI (Destination Access Point Identifier) do not match the received SM SAPI and/or DAPI for at least 3 consecutive TTI (Trail Trace Identifier) of the 256 frames multiframe.</p> <p>OBDI: OTU - Backward Defect Indication (OTU-BDI) when the BDI (Backward Defect Indication) bit in the SM overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OIAE: OTU - Incoming Alignment Error (OTU-IAE) when IAE bit in the SM overhead field (byte 3, bit 6) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OBIAe: OTU - Backward Incoming Alignment Error (OTU-BIAE) when BEI (Backward Error Indication) /BIAE (Backward Incoming Alignment Error) bits in the SM overhead field (byte 3, bits 1 to 4) are ""1011"" for at least 3 consecutive frames.</p>
---------------------	---

**Response Syntax** <History>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:E[1..n]:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of Optical Transport Unit (OTU) alarm. PRESENT, indicates that at least one alarm has occurred. ABSENT, indicates that no alarm occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:OTU3:E1:HIST? OAIS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:E[1..n]:SECnds?

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:E[1..n]:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which Optical Transport Unit (OTU) alarm occurred for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU3e(1/2)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:E[1..n]:SEConds? &lt;wsp&gt;LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIAe</p>

---

## SCPI Command Reference

### Alarms/Errors

---

:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:E[1..n]:SEConds?

Parameter(s)	Alarm:
	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIae</p> <p>Selects the Optical Transport Unit (OTU) alarm type.</p> <p>LOF: Loss Of Frame (LOF) when OOF is present for at least 3 ms.</p> <p>OOF: Out-Of-Frame (OOF) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU (Optical Transport Unit) frames.</p> <p>LOM: Loss Of Multiframe (LOM) when OOM (Out of Multiframe) is present for at least 3 ms.</p> <p>OOM: Out-Of-Multiframe (OOM) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.</p> <p>OAIS: OTU - Alarm Indication Signal (OTU-AIS) when polynomial number 11 (PN-11) is over all OTU (Optical Transport Unit) frame bits including FAS and MFAS (Multiframe Alignment Signal) for at least 3 consecutive 8192 bit-interval.</p> <p>OTIM: OTU - Trace Identifier Mismatch (OTU-TIM) when expected SM SAPI (Source Access Point Identifier) and/or SM DAPI (Destination Access Point Identifier) do not match the received SM SAPI and/or DAPI for at least 3 consecutive TTI (Trail Trace Identifier) of the 256 frames multiframe.</p> <p>OBDI: OTU - Backward Defect Indication (OTU-BDI) when the BDI (Backward Defect Indication) bit in the SM overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OIAE: OTU - Incoming Alignment Error (OTU-IAE) when IAE bit in the SM overhead field (byte 3, bit 6) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OBIae: OTU - Backward Incoming Alignment Error (OTU-BIAE) when BEI (Backward Error Indication) /BIAE (Backward Incoming Alignment Error) bits in the SM overhead field (byte 3, bits 1 to 4) are ""1011"" for at least 3 consecutive frames.</p>

**Response Syntax** <Seconds>

---



---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:E[1..n]:SEConds?**

---

<b>Response(s)</b>	<b>Seconds:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Transport Unit (OTU) alarm.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:OTU3:E1:SEC? OAIS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:E[1..n]:HISTory?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:CURRent?

---

<b>Description</b>	<p>This query returns the current status of Optical Transport Unit (OTU) alarm for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU3e(1/2)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:CURRent? &lt;wsp&gt;LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIAe</p>

---

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OTU[1..n]:E[1..n]:CURRent?**

---

<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF   OOF   LOM   OOM   OAIS   OTIM   OBDI   OIAE   OBIae</p> <p>Selects the Optical Transport Unit (OTU) alarm type.</p> <p>LOF: Loss Of Frame (LOF) when OOF is present for at least 3 ms.</p> <p>OOF: Out-Of-Frame (OOF) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU (Optical Transport Unit) frames.</p> <p>LOM: Loss Of Multiframe (LOM) when OOM (Out of Multiframe) is present for at least 3 ms.</p> <p>OOM: Out-Of-Multiframe (OOM) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.</p> <p>OAIS: OTU - Alarm Indication Signal (OTU-AIS) when polynomial number 11 (PN-11) is over all OTU (Optical Transport Unit) frame bits including FAS and MFAS (Multiframe Alignment Signal) for at least 3 consecutive 8192 bit-interval.</p> <p>OTIM: OTU - Trace Identifier Mismatch (OTU-TIM) when expected SM SAPI (Source Access Point Identifier) and/or SM DAPI (Destination Access Point Identifier) do not match the received SM SAPI and/or DAPI for at least 3 consecutive TTI (Trail Trace Identifier) of the 256 frames multiframe.</p> <p>OBDI: OTU - Backward Defect Indication (OTU-BDI) when the BDI (Backward Defect Indication) bit in the SM overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OIAE: OTU - Incoming Alignment Error (OTU-IAE) when IAE bit in the SM overhead field (byte 3, bit 6) is ""1"" for at least 5 consecutive OTU frames.</p> <p>OBIae: OTU - Backward Incoming Alignment Error (OTU-BIAE) when BEI (Backward Error Indication) /BIAE (Backward Incoming Alignment Error) bits in the SM overhead field (byte 3, bits 1 to 4) are ""1011"" for at least 3 consecutive frames.</p>
---------------------	---

**Response Syntax** <Current>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of Optical Transport Unit (OTU) alarm. PRESENT, indicates that at least one alarm has occurred in the last second. ABSENT, indicates that there is no alarm. INACTIVE, indicates that the test is not running.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:OTU3:E1:CURR? OAIS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:HISTory?

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:HISTory?**

<b>Description</b>	<p>This query returns the history status of the Optical Transport Unit (OTU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:HISTory? <wsp>OBIP8   OBEI   FAS   MFAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the type of OTU (Optical Transport Unit) error.</p> <p>OBIP8: the type of OTU-BIP-8 (OTU - Bit Interleave Parity-8) error.</p> <p>OBEI: the type of OTU-BEI (OTU - Backward Error Indication) error.</p> <p>FAS: the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS: the type of MFAS (Multiframe Alignment Signal) error.</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of Optical Transport Unit (OTU) error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:OTU1:HIST? OBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which the Optical Transport Unit (OTU) error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:SEConds? &lt;wsp&gt;OBIP8   OBEI   FAS   MFAS</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the type of OTU (Optical Transport Unit) error.</p> <p>OBIP8: the type of OTU-BIP-8 (OTU - Bit Interleave Parity-8) error.</p> <p>OBEI: the type of OTU-BEI (OTU - Backward Error Indication) error.</p> <p>FAS: the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS: the type of MFAS (Multiframe Alignment Signal) error.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ERR:OTU1:SEC? OBIP8</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:CURRent?</p>

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:CURRent?**

<b>Description</b>	<p>This query returns the current status of Optical Transport Unit (OTU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:CURRent? <wsp>OBIP8   OBEI   FAS   MFAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the type of OTU (Optical Transport Unit) error.</p> <p>OBIP8: the type of OTU-BIP-8 (OTU - Bit Interleave Parity-8) error.</p> <p>OBEI: the type of OTU-BEI (OTU - Backward Error Indication) error.</p> <p>FAS: the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS: the type of MFAS (Multiframe Alignment Signal) error.</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of the Optical Transport Unit (OTU) error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:OTU1:CURR? OBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:SECConds?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:OTN:ERRor:OTU[1..n]:COUNt?**

<b>Description</b>	<p>This query returns the count of the Optical Transport Unit (OTU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:OTN:ERRor:OTU[1..n]:COUNt? &lt;wsp&gt;OBIP8   OBEI   FAS   MFAS</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the type of OTU (Optical Transport Unit) error.</p> <p>OBIP8: the type of OTU-BIP-8 (OTU - Bit Interleave Parity-8) error.</p> <p>OBEI: the type of OTU-BEI (OTU - Backward Error Indication) error.</p> <p>FAS: the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS: the type of MFAS (Multiframe Alignment Signal) error.</p>
<b>Response Syntax</b>	<p>&lt;Count&gt;</p>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of the Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ERR:OTU1:COUN? OBIP8</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:ERRor:OTU[1..n]:CURRent?</p>



**:FETCh[1..n]:DATA:TELeom:OTN:ERRor:OTU[1..n]:RATE?**

<b>Description</b>	<p>This query returns the current rate of Optical Transport Unit (OTU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:OTN:ERRor:OTU[1..n]:RATE? <wsp>OBIP8   OBEI   FAS   MFAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the type of OTU (Optical Transport Unit) error.</p> <p>OBIP8: the type of OTU-BIP-8 (OTU - Bit Interleave Parity-8) error.</p> <p>OBEI: the type of OTU-BEI (OTU - Backward Error Indication) error.</p> <p>FAS: the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS: the type of MFAS (Multiframe Alignment Signal) error.</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of the Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:OTU1:RATE? OBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:OTN:ERRor:OTU[1..n]:COUNT?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:OTN:ERRor:OTU[1..n]:E[1..n]:HISTory?

---

<b>Description</b>	<p>This query returns the history status of Optical Transport Unit (OTU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OUT3e(1/2)</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELecom:OTN:ERRor:OTU[1..n]:E[1..n]:HISTory? &lt;wsp&gt;OBIP8   OBEI   FAS   MFAS</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8: OTU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI: OTU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS: Frame Alignment Signal (FAS) error.</p> <p>MFAS: Multiframe Alignment Signal (MFAS) error.</p>
<b>Response Syntax</b>	<code>&lt;History&gt;</code>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of Optical Transport Unit (OTU) error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AMO 15 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:INJ FETC:DATA:TEL:OTN:ERR:OTU3:E1:HIST? OBIP8</pre>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELecom:OTN:ERRor:OTU[1..n]:E[1..n]:COUNT?</code>

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which Optical Transport Unit (OTU) error occurred for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OUT3e(1/2)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:SEConds? <wsp>OBIP8   OBEI   FAS   MFAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8: OTU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI: OTU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS: Frame Alignment Signal (FAS) error.</p> <p>MFAS: Multiframe Alignment Signal (MFAS) error.</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:OTU3:E1:SEC? OBIP8</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:RATE?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:CURRent?

---

<b>Description</b>	<p>This query returns the current status of Optical Transport Unit (OTU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OUT3e(1/2)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:CURRent? <wsp>OBIP8   OBEI   FAS   MFAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8: OTU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI: OTU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS: Frame Alignment Signal (FAS) error.</p> <p>MFAS: Multiframe Alignment Signal (MFAS) error.</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of Optical Transport Unit (OTU) error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test is not running.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:OTU3:E1:CURR? OBIP8</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:SECnds?

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:COUNT?**

<b>Description</b>	<p>This query returns the count of Optical Transport Unit (OTU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OUT3e(1/2)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:COUNT? <wsp>OBIP8   OBEI   FAS   MFAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8: OTU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI: OTU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS: Frame Alignment Signal (FAS) error.</p> <p>MFAS: Multiframe Alignment Signal (MFAS) error.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:OTU3:E1:COUN? OBIP8</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:RATE?

---

<b>Description</b>	<p>This query returns the current rate of Optical Transport Unit (OTU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU3e(1/2)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:RATE? &lt;wsp&gt;OBIP8   OBEI   FAS   MFAS</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8: OTU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI: OTU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS: Frame Alignment Signal (FAS) error.</p> <p>MFAS: Multiframe Alignment Signal (MFAS) error.</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:OTU3:E1:RATE? OBIP8</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:CURRent?</p>

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:HISTory?**

---

<b>Description</b>	<p>This query returns the history status of the Optical Data Unit (ODU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:HISTory? &lt;wsp&gt;OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OTIM   OFSD   OBSD   LOFLom[, &lt;Channel&gt;]</p>

---

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:HISTory?

Parameter(s)	Alarm:
	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p>
	<p>The allowed elements for this parameter are: OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OTIM   OFSD   OBSD   LOFLom</p>
	<p>Selects the type of Optical Data Unit (ODU) alarm.</p>
	<p>OAIS, OAIS (ODU - Alarm Indication Signal) indicates that the STAT information detected, byte 3, bits 6 to 8 is ""111"" for at least 3 consecutive frames.</p>
	<p>OBDI, ODU-BDI (ODU - Backward Defect indication) is declared when the BDI (Backward Defect Indication) bit in the PM (Performance Monitoring) overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive frames.</p>
	<p>OLCK, OLCK (ODU - Lock) indicates that the STAT information detected is ""101"" for at least 3 consecutive frames.</p>
	<p>OOCI, OOCI (ODU - Open Connection Indication) indicates that the STAT information detected is ""110"" for at least 3 consecutive frames.</p>
	<p>OSFS, ODU-FSF (ODU - Forward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""00000001"".</p>
	<p>OBSF, ODU-BSF (ODU - Backward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""00000001"".</p>
	<p>OTIM, ODU-TIM (ODU - Trace Identification Mismatch) is declared when the received SAPI (Source Access Point Identifier) and/or DAPI (Destination Access Point Identifier) do not match the expected SAPI and/or DAPI. This alarm is only available when TIM SAPI or DAPI is enabled.</p>
	<p>OFSD, ODU-FSD (ODU - Forward Signal Degrade) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""00000010"" OBSD, ODU-BSD (ODU - Backward Signal Degrade) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""00000010"".</p>
	<p>LOFLom, ODU-LOFLom (ODU-Loss of Frame Loss of Multiframe) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.</p>



---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:HISTory?**

<b>Parameter(s)</b>	<b>Channel:</b> The program data syntax for the second parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the channel that will be used by command/query. This parameter can only be used in Multi-Channel OTN. The numeric channel ranges from [1:n] in function of ODU Mapping.
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of the Optical Data Unit (ODU) alarm. PRESENT, indicates that at least one alarm has occurred. ABSENT, indicates that no alarm occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:ODU1:HIST? OAIS FETC:DATA:TEL:OTN:ALAR:ODU100:HIST? OAIS, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:SEConds?

---

## SCPI Command Reference

### *Alarms/Errors*

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:SEConds?**

---

<b>Description</b>	<p>This query returns the number of seconds within which the Optical Data Unit (ODU) alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:SEConds? &lt;wsp&gt;O AIS   OBDI   OLCK   OOCI   OFSF   OBSF   OTIM   OFSD   OBSD   LOFLom[, &lt;Channel&gt;]</p>

---

**:FETCh[1..n]:DATA:TELeom:OTN:ALARm:ODU[1..n]:SEConDs?****Parameter(s)****Alarm:**

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: OAIS | OBDI | OLCK | OOCI | OFSF | OBSF | OTIM | OFSD | OBSD | LOFLom

Selects the type of Optical Data Unit (ODU) alarm.

OAIS, OAIS (ODU - Alarm Indication Signal) indicates that the STAT information detected, byte 3, bits 6 to 8 is ""111"" for at least 3 consecutive frames.

OBDI, ODU-BDI (ODU - Backward Defect indication) is declared when the BDI (Backward Defect Indication) bit in the PM (Performance Monitoring) overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive frames.

OLCK, OLCK (ODU - Lock) indicates that the STAT information detected is ""101"" for at least 3 consecutive frames.

OOCI, OOCI (ODU - Open Connection Indication) indicates that the STAT information detected is ""110"" for at least 3 consecutive frames.

OSF, ODU-FSF (ODU - Forward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""0000001"".

OBSF, ODU-BSF (ODU - Backward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""0000001"".

OTIM, ODU-TIM (ODU - Trace Identification Mismatch) is declared when the received SAPI (Source Access Point Identifier) and/or DAPI (Destination Access Point Identifier) do not match the expected SAPI and/or DAPI. This alarm is only available when TIM SAPI or DAPI is enabled.

OFSD, ODU-FSD (ODU - Forward Signal Degrade) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""0000010"" OBSD, ODU-BSD (ODU - Backward Signal Degrade) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""0000010"".

OLOFLom, ODU-LOFLOM (ODU-Loss of Frame Loss of Multiframe) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:SEConds?

---

<b>Parameter(s)</b>	<b>Channel:</b> The program data syntax for the second parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the channel that will be used by command/query. This parameter can only be used in Multi-Channel OTN. The numeric channel ranges from [1:n] in function of ODU Mapping.
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<b>Seconds:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of the Optical Data Unit (ODU) alarm.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:ODU1:SEC? OAIS FETC:DATA:TEL:OTN:ALAR:ODU100:SEC? OAIS, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:HISTory?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CURRent?**

---

**Description**

This query returns the current status of the Optical Data Unit (ODU) alarm.

At \*RST condition, this value is set to device-dependent.

Navigation Path: Test > OTN BERT > Results > Alarms/Errors > ODU

**Syntax**

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CURRent? <wsp>OAIS | OBDI |  
OLCK | OOCI | OFSF | OBSF | OTIM | OFSD | OBSD | LOFLom[, <Channel>]

---

**:FETCh[1..n]:DATA:TELeom:OTN:ALARm:ODU[1..n]:CURRent?**

Parameter(s)	Alarm:
	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p>
	<p>The allowed elements for this parameter are: OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OTIM   OFSD   OBSD   LOFLom</p>
	<p>Selects the type of Optical Data Unit (ODU) alarm.</p>
	<p>OAIS, OAIS (ODU - Alarm Indication Signal) indicates that the STAT information detected, byte 3, bits 6 to 8 is ""111"" for at least 3 consecutive frames.</p>
	<p>OBDI, ODU-BDI (ODU - Backward Defect indication) is declared when the BDI (Backward Defect Indication) bit in the PM (Performance Monitoring) overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive frames.</p>
	<p>OLCK, OLCK (ODU - Lock) indicates that the STAT information detected is ""101"" for at least 3 consecutive frames.</p>
	<p>OOCI, OOCI (ODU - Open Connection Indication) indicates that the STAT information detected is ""110"" for at least 3 consecutive frames.</p>
	<p>OSFS, ODU-FSF (ODU - Forward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""00000001"".</p>
	<p>OBSF, ODU-BSF (ODU - Backward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""00000001"".</p>
	<p>OTIM, ODU-TIM (ODU - Trace Identification Mismatch) is declared when the received SAPI (Source Access Point Identifier) and/or DAPI (Destination Access Point Identifier) do not match the expected SAPI and/or DAPI. This alarm is only available when TIM SAPI or DAPI is enabled.</p>
	<p>OFSD, ODU-FSD (ODU - Forward Signal Degrade) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""00000010""</p>
	<p>OBSD, ODU-BSD (ODU - Backward Signal Degrade) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""00000010"".</p>
	<p>OLOFLom, ODU-LOFLOM (ODU-Loss of Frame Loss of Multiframe) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.</p>

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:CURRent?**

<b>Parameter(s)</b>	<b>Channel:</b> The program data syntax for the second parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the channel that will be used by command/query. This parameter can only be used in Multi-Channel OTN. The numeric channel ranges from [1:n] in function of ODU Mapping.
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of the Optical Data Unit (ODU) alarm. PRESENT, indicates that at least one alarm has occurred in the last second. ABSENT, indicates that there is no alarm. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:ODU1:CURR? OAIS FETC:DATA:TEL:OTN:ALAR:ODU100:CURR? OAIS, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:SEConds?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:HISTory?

---

<b>Description</b>	<p>This query returns the history status of Optical Data Unit (ODU) alarm for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU(1/2)e</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:HISTory? &lt;wsp&gt;OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OTIM   OFSD   OBSD   LOFLom</p>

---



---

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:HISTory?

---

<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OTIM   OFSD   OBSD   LOFLom</p> <p>Selects the Optical Data Unit (ODU) alarm type.</p> <p>OAIS: ODU - Alarm Indication Signal (OAIS) indicates that the STAT information detected, byte 3, bits 6 to 8 is ""111"" for at least 3 consecutive frames.</p> <p>OBDI: ODU - Backward Defect indication (ODU-BDI) is declared when the BDI (Backward Defect Indication) bit in the PM (Performance Monitoring) overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive frames.</p> <p>OLCK: ODU - Lock (OLCK) indicates that the STAT information detected is ""101"" for at least 3 consecutive frames.</p> <p>OOCI: ODU - Open Connection Indication (OOCI) indicates that the STAT information detected is ""110"" for at least 3 consecutive frames.</p> <p>OSFS: ODU - Forward Signal Fail (ODU-FSF) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""00000001"".</p> <p>OBSF: ODU - Backward Signal Fail (ODU-BSF) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""00000001"".</p> <p>OTIM: ODU - Trace Identification Mismatch (ODU-TIM) is declared when the received SAPI (Source Access Point Identifier) and/or DAPI (Destination Access Point Identifier) do not math the expected SAPI and/or DAPI. This alarm is only available when TIM SAPI or DAPI is enabled.</p> <p>OFSD: ODU - Forward Signal Degrade (ODU-FSD) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""00000010""</p> <p>OBSD: ODU - Backward Signal Degrade (ODU-BSD) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""00000010"".</p> <p>LOFLom, ODU-LOFLOM (ODU-Loss of Frame Loss of Multiframe) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.</p>
---------------------	--

**Response Syntax** <History>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of Optical Data Unit (ODU) alarm. PRESENT, indicates that at least one alarm has occurred. ABSENT, indicates that no alarm occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:ODU3:E1:HIST? OAIS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:SEConds?

---

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:SEConds?**

---

**Description**

This query returns the number of seconds within which Optical Data Unit (ODU) alarm occurred for overclocked rates OTU3e1/2.

At \*RST condition, this value is device dependent.

Navigation Path: Test > OTN BERT > Results > Alarms/Errors > ODU3e(1/2)

Navigation Path: Test > OTN BERT > Results > Alarms/Errors > ODU(1/2)e

**Syntax**

:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:SEConds? <wsp>OAIS | OBDI | OLCK | OOCI | OFSF | OBSF | OTIM | OFSD | OBSD | LOFLom

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:SEConds?**

<b>Parameter(s)</b>	<b>Alarm:</b>
	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OTIM   OFSD   OBSD   LOFLom</p> <p>Selects the Optical Data Unit (ODU) alarm type.</p> <p>OAIS: ODU - Alarm Indication Signal (OAIS) indicates that the STAT information detected, byte 3, bits 6 to 8 is ""111"" for at least 3 consecutive frames.</p> <p>OBDI: ODU - Backward Defect indication (ODU-BDI is declared when the BDI (Backward Defect Indication) bit in the PM (Performance Monitoring) overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive frames.</p> <p>OLCK: ODU - Lock (OLCK) indicates that the STAT information detected is ""101"" for at least 3 consecutive frames.</p> <p>OOCI: ODU - Open Connection Indication (OOCI) indicates that the STAT information detected is ""110"" for at least 3 consecutive frames.</p> <p>OSFS: ODU - Forward Signal Fail (ODU-FSF) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""00000001"".</p> <p>OBSF: ODU - Backward Signal Fail (ODU-BSF) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""00000001"".</p> <p>OTIM: ODU - Trace Identification Mismatch (ODU-TIM) is declared when the received SAPI (Source Access Point Identifier) and/or DAPI (Destination Access Point Identifier) do not math the expected SAPI and/or DAPI. This alarm is only available when TIM SAPI or DAPI is enabled.</p> <p>OFSD: ODU - Forward Signal Degrade (ODU-FSD) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""00000010""</p> <p>OBSD: ODU - Backward Signal Degrade (ODU-BSD) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""00000010"".</p> <p>LOFLom, ODU-LOFLOM (ODU-Loss of Frame Loss of Multiframe) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.</p>

**Response Syntax** <Seconds>

---

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:SEConds?**

---

<b>Response(s)</b>	<b>Seconds:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Data Unit (ODU) alarm.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:ODU3:E1:SEC? OAIS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:HISTory?

---

## SCPI Command Reference

### Alarms/Errors

---

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:CURRent?**

---

<b>Description</b>	<p>This query returns the current status of Optical Data Unit (ODU) alarm for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU(1/2)e</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:CURRent? &lt;wsp&gt;OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OTIM   OFSD   OBSD   LOFLom</p>

---

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:CURRent?**

<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OTIM   OFSD   OBSD   LOFLom</p> <p>Selects the Optical Data Unit (ODU) alarm type.</p> <p>OAIS: ODU - Alarm Indication Signal (OAIS) indicates that the STAT information detected, byte 3, bits 6 to 8 is ""111"" for at least 3 consecutive frames.</p> <p>OBDI: ODU - Backward Defect indication (ODU-BDI is declared when the BDI (Backward Defect Indication) bit in the PM (Performance Monitoring) overhead field (byte 3, bit 5) is ""1"" for at least 5 consecutive frames.</p> <p>OLCK: ODU - Lock (OLCK) indicates that the STAT information detected is ""101"" for at least 3 consecutive frames.</p> <p>OOCI: ODU - Open Connection Indication (OOCI) indicates that the STAT information detected is ""110"" for at least 3 consecutive frames.</p> <p>OSFS: ODU - Forward Signal Fail (ODU-FSF) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""00000001"".</p> <p>OBSF: ODU - Backward Signal Fail (ODU-BSF) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""00000001"".</p> <p>OTIM: ODU - Trace Identification Mismatch (ODU-TIM) is declared when the received SAPI (Source Access Point Identifier) and/or DAPI (Destination Access Point Identifier) do not math the expected SAPI and/or DAPI. This alarm is only available when TIM SAPI or DAPI is enabled.</p> <p>OFSD: ODU - Forward Signal Degrade (ODU-FSD) is declared when the received FTFL (Fault Type Fault Location) byte 0 is ""00000010""</p> <p>OBSD: ODU - Backward Signal Degrade (ODU-BSD) is declared when the received FTFL (Fault Type Fault Location) byte 128 is ""00000010"".</p> <p>LOFLom, ODU-LOFLOM (ODU-Loss of Frame Loss of Multiframe) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.</p>
---------------------	---

**Response Syntax** <Current>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of Optical Data Unit (ODU) alarm. PRESENT, indicates that at least one alarm has occurred in the last second. ABSENT, indicates that there is no alarm. INACTIVE, indicates that the test is not running.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:ODU3:E1:CURR? OAIS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:SEConds?

---



**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:HISTory?**

<b>Description</b>	<p>This query returns the history status of the Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:HISTory? &lt;wsp&gt;OBIP8   OBEI[, &lt;Channel&gt;]</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and the locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<History>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of Optical Data Unit (ODU) error. PRESENT, indicates that at least one error has occurred. ABSENT, indicates that no error occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:ODU1:HIST? OBIP8 FETC:DATA:TEL:OTN:ERR:ODU100:HIST? OBIP8, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:SEConds?

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the Optical Data Unit (ODU) error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:SEConds? &lt;wsp&gt;OBIP8   OBEI[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and the locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:SEConds?**

<b>Response(s)</b>	<b>Seconds:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of the Optical Data Unit (ODU) error.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:ODU1:SEC? OBIP8 FETC:DATA:TEL:OTN:ERR:ODU100:SEC? OBIP8, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:HISTory?

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CURRent?**

<b>Description</b>	<p>This query returns the current status of the Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CURRent? <wsp>OBIP8   OBEI[, <Channel>]
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and the locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Current>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of the Optical Data Unit (ODU) error. PRESENT, indicates that at least one error has occurred in the last second. ABSENT, indicates that there is no error. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:ODU1:CURR? OBIP8 FETC:DATA:TEL:OTN:ERR:ODU100:CURR? OBIP8, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:COUNT?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:COUNT?**

<b>Description</b>	<p>This query returns the count of Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:COUNT? &lt;wsp&gt;OBIP8   OBEI[, &lt;Channel&gt;]</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and the locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<code>&lt;Count&gt;</code>

---

## SCPI Command Reference

### Alarms/Errors

---

<b>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:COUNT?</b>	
<b>Response(s)</b>	<b>Count:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of the Optical Data Unit (ODU) error.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:ODU1:COUN? OBIP8 FETC:DATA:TEL:OTN:ERR:ODU100:COUN? OBIP8, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:RATE?

---



**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:RATE?**

<b>Description</b>	<p>This query returns the current rate of the Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:RATE? <wsp>OBIP8   OBEI[, <Channel>]
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and the locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of the Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ERR:ODU1:RATE? OBIP8</p> <p>FETC:DATA:TEL:OTN:ERR:ODU100:RATE? OBIP8, 3</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:COUNT?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELeCom:OTN:ERRor:ODU[1..n]:E[1..n]:HISTory?**

<b>Description</b>	<p>This query returns the history status of Optical Data Unit (ODU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU(1/2)e</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELeCom:OTN:ERRor:ODU[1..n]:E[1..n]:HISTory? &lt;wsp&gt;OBIP8   OBEI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8: the ODU - Bit Interleave Parity-8 (ODU-BIP-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU - Backward Error Indication (ODU-BEI) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of Optical Data Unit (ODU) error. PRESENT, indicates that at least one error has occurred. ABSENT, indicates that no error occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ FETC:DATA:TEL:OTN:ERR:ODU3:E1:HIST? OBIP8
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUNT SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:INJect

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which Optical Data Unit (ODU) error occurred for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU(1/2)e</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:SEConds? &lt;wsp&gt;OBIP8   OBEI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8: the ODU - Bit Interleave Parity-8 (ODU-BIP-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU - Backward Error Indication (ODU-BEI) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:ODU3:E1:SEC? OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:INJECT</p>

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:CURRent?**

<b>Description</b>	<p>This query returns the current status of Optical Data Unit (ODU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU(1/2)e</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:CURRent? &lt;wsp&gt;OBIP8   OBEI</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8: the ODU - Bit Interleave Parity-8 (ODU-BIP-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU - Backward Error Indication (ODU-BEI) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Response Syntax</b>	<code>&lt;Current&gt;</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of Optical Data Unit (ODU) error. PRESENT, indicates that at least one error has occurred in the last second. ABSENT, indicates that there is no error. INACTIVE, indicates that the test is not running.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ FETC:DATA:TEL:OTN:ERR:ODU3:E1:CURR? OBIP8
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUNT SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:INJect

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:COUNT?**

<b>Description</b>	<p>This query returns the count of Optical Data Unit (ODU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU(1/2)e</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:COUNT? <wsp>OBIP8   OBEI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8: the ODU - Bit Interleave Parity-8 (ODU-BIP-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU - Backward Error Indication (ODU-BEI) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:ODU3:E1:COUN? OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:INJect</p>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:RATE?**

<b>Description</b>	<p>This query returns the current rate of Optical Data Unit (ODU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU(1/2)e</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:RATE? &lt;wsp&gt;OBIP8   OBEI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8: the ODU - Bit Interleave Parity-8 (ODU-BIP-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU - Backward Error Indication (ODU-BEI) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:ODU3:E1:RATE? OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:INJECT</p>



---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OPU[1..n]:HISTory?**

---

<b>Description</b>	<p>This query returns the history status of the Optical Payload Unit (OPU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; OPU</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OPU[1..n]:HISTory? &lt;wsp&gt;OAIS   OPLM   OMSim   OCSF   LOOMFI   OOMFI[, &lt;Channel&gt;]</code>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OPLM   OMSim   OCSF   LOOMFI   OOMFI</p> <p>Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OAIS: OPU-AIS (Optical Payload Unit Alarm Indication Signal)</p> <p>OPLM: OPU-PLM (Optical Payload Unit Payload Mismatch)</p> <p>OMSim: OPU-MSIM (Optical Payload Unit Multiplex Structure Identifier Mismatch)</p> <p>OCSF: OPU-CSF (Optical Payload Unit Client Signal Fail)</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<code>&lt;History&gt;</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of the Optical Payload Unit (OPU) alarm. PRESENT, indicates that at least one alarm has occurred. ABSENT, indicates that no alarm occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:OPU1:HIST? OPLM FETC:DATA:TEL:OTN:ALAR:OPU100:HIST? OPLM, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CURRent?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the Optical Payload Unit (OPU) alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; OPU</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:SEConds? &lt;wsp&gt;OAIS   OPLM   OMSim   OCSF   LOOMFI   OOMFI[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OPLM   OMSim   OCSF   LOOMFI   OOMFI</p> <p>Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OAIS: OPU-AIS (Optical Payload Unit Alarm Indication Signal)</p> <p>OPLM: OPU-PLM (Optical Payload Unit Payload Mismatch)</p> <p>OMSim: OPU-MSIM (Optical Payload Unit Multiplex Structure Identifier Mismatch)</p> <p>OCSF: OPU-CSF (Optical Payload Unit Client Signal Fail)</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:SEConds?**

<b>Response(s)</b>	<b>Seconds:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of the Optical Payload Unit (OPU) alarm.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:OPU1:SEC? OPLM FETC:DATA:TEL:OTN:ALAR:OPU100:SEC? OPLM, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CURRent?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CURRent?**

---

<b>Description</b>	<p>This query returns the current status of the Optical Payload Unit (OPU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; OPU</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CURRent? &lt;wsp&gt;OAIS   OPLM   OMSim   OCSF   LOOMFI   OOMFI[, &lt;Channel&gt;]</code>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OPLM   OMSim   OCSF   LOOMFI   OOMFI</p> <p>Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OAIS: OPU-AIS (Optical Payload Unit Alarm Indication Signal)</p> <p>OPLM: OPU-PLM (Optical Payload Unit Payload Mismatch)</p> <p>OMSim: OPU-MSIM (Optical Payload Unit Multiplex Structure Identifier Mismatch)</p> <p>OCSF: OPU-CSF (Optical Payload Unit Client Signal Fail)</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<code>&lt;Current&gt;</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OPU[1..n]:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of the Optical Payload Unit (OPU) alarm. PRESENT, indicates that at least one alarm has occurred in the last second. ABSENT, indicates that there is no alarm. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:OPU1:CURR? OPLM FETC:DATA:TEL:OTN:ALAR:OPU100:CURR? OPLM, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ALARm:OPU[1..n]:HISTory?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:HISTory?**

---

<b>Description</b>	<p>This query returns the history status of Forward Error Correction (FEC) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU &gt; Errors &gt; FEC</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:HISTory? &lt;wsp&gt;FCCW   FUCW   FCSYmb   FCBit</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCCW   FUCW   FCSYmb   FCBit</p> <p>Selects the type of FEC (Forward Error Correction) error.</p> <p>FCCW: the FEC-CORR-CW (Forward Error Correction - Correctable - Codeword) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW: the FEC-UNCORR-CW (Forward Error Correction - Uncorrectable - Codeword) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.</p> <p>FCSYmb: the FEC-CORR-SYMB (Forward Error Correction - Correctable - Symbol) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit: the FEC-CORR-BIT (Forward Error Correction - Correctable - Bit) which generates 1 symbol (byte) containing 1 bit in error.</p>
<b>Response Syntax</b>	<code>&lt;History&gt;</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of Forward Error Correction (FEC) error. PRESENT, indicates that at least one error has occurred. ABSENT, indicates that no error occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE FCCW SOUR:DATA:TEL:OTN:ERR:FEC:AMO 15 SOUR:DATA:TEL:OTN:ERR:FEC:INJ FETC:DATA:TEL:OTN:ERR:FEC:HIST? FCCW
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:CURRent?

---



**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the Forward Error Correction (FEC) error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU &gt; Errors &gt; FEC</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:SEConds? <wsp>FCCW   FUCW   FCSYmb   FCBit
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCCW   FUCW   FCSYmb   FCBit</p> <p>Selects the type of FEC (Forward Error Correction) error.</p> <p>FCCW: the FEC-CORR-CW (Forward Error Correction - Correctable - Codeword) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW: the FEC-UNCORR-CW (Forward Error Correction - Uncorrectable - Codeword) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.</p> <p>FCSYmb: the FEC-CORR-SYMB (Forward Error Correction - Correctable - Symbol) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit: the FEC-CORR-BIT (Forward Error Correction - Correctable - Bit) which generates 1 symbol (byte) containing 1 bit in error.</p>
<b>Response Syntax</b>	<Seconds>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:SEConds?**

<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of Forward Error Correction (FEC) error.</p> <p>FCCW, selects the FEC-CORR-CW (Forward Error Correction - Correctable - Codeword) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW, selects the FEC-UNCORR-CW (Forward Error Correction - Uncorrectable - Codeword) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.</p> <p>FCSYmb, selects the FEC-CORR-SYMB (Forward Error Correction - Correctable - Symbol) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit, selects the FEC-CORR-BIT (Forward Error Correction - Correctable - Bit) which generates 1 symbol (byte) containing 1 bit in error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:FEC:SEC? FCCW</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:CURRent?</p>

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:CURRent?**

<b>Description</b>	<p>This query returns the current status of Forward Error Correction (FEC) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU &gt; Errors &gt; FEC</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:CURRent? &lt;wsp&gt;FCCW   FUCW   FCSYmb   FCBit</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCCW   FUCW   FCSYmb   FCBit</p> <p>Selects the type of FEC (Forward Error Correction) error.</p> <p>FCCW: the FEC-CORR-CW (Forward Error Correction - Correctable - Codeword) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW: the FEC-UNCORR-CW (Forward Error Correction - Uncorrectable - Codeword) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.</p> <p>FCSYmb: the FEC-CORR-SYMB (Forward Error Correction - Correctable - Symbol) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit: the FEC-CORR-BIT (Forward Error Correction - Correctable - Bit) which generates 1 symbol (byte) containing 1 bit in error.</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of Forward Error Correction (FEC) error. PRESENT, indicates that at least one error has occurred in the last second. ABSENT, indicates that there is no error. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE FCCW SOUR:DATA:TEL:OTN:ERR:FEC:AMO 15 SOUR:DATA:TEL:OTN:ERR:FEC:INJ FETC:DATA:TEL:OTN:ERR:FEC:CURR? FCCW
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:HISTory?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:COUNT?**

---

<b>Description</b>	<p>This query returns the count of Forward Error Correction (FEC) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU &gt; Errors &gt; FEC</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:COUNT? &lt;wsp&gt;FCCW   FUCW   FCSYmb   FCBit</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCCW   FUCW   FCSYmb   FCBit</p> <p>Selects the type of FEC (Forward Error Correction) error.</p> <p>FCCW: the FEC-CORR-CW (Forward Error Correction - Correctable - Codeword) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW: the FEC-UNCORR-CW (Forward Error Correction - Uncorrectable - Codeword) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.</p> <p>FCSYmb: the FEC-CORR-SYMB (Forward Error Correction - Correctable - Symbol) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit: the FEC-CORR-BIT (Forward Error Correction - Correctable - Bit) which generates 1 symbol (byte) containing 1 bit in error.</p>
<b>Response Syntax</b>	<code>&lt;Count&gt;</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:COUNT?**

<b>Response(s)</b>	<b>Count:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of Forward Error Correction (FEC) error. FCCW, selects the FEC-CORR-CW (Forward Error Correction - Correctable - Codeword) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword. FUCW, selects the FEC-UNCORR-CW (Forward Error Correction - Uncorrectable - Codeword) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword. FCSYmb, selects the FEC-CORR-SYMB (Forward Error Correction - Correctable - Symbol) which generates 1 symbol (byte) containing 8 bits in error. FCBIt, selects the FEC-CORR-BIT (Forward Error Correction - Correctable - Bit) which generates 1 symbol (byte) containing 1 bit in error.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE FCCW SOUR:DATA:TEL:OTN:ERR:FEC:AMO 15 SOUR:DATA:TEL:OTN:ERR:FEC:INJ FETC:DATA:TEL:OTN:ERR:FEC:COUN? FCCW
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:CURRent?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:RATE?**

<b>Description</b>	<p>This query returns the current rate of Forward Error Correction (FEC) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; OTU &gt; Errors &gt; FEC</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:RATE? &lt;wsp&gt;FCCW   FUCW   FCSYmb   FCBit</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCCW   FUCW   FCSYmb   FCBit</p> <p>Selects the type of FEC (Forward Error Correction) error.</p> <p>FCCW: the FEC-CORR-CW (Forward Error Correction - Correctable - Codeword) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW: the FEC-UNCORR-CW (Forward Error Correction - Uncorrectable - Codeword) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.</p> <p>FCSYmb: the FEC-CORR-SYMB (Forward Error Correction - Correctable - Symbol) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit: the FEC-CORR-BIT (Forward Error Correction - Correctable - Bit) which generates 1 symbol (byte) containing 1 bit in error.</p>
<b>Response Syntax</b>	<code>&lt;Rate&gt;</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:RATE?**

<b>Response(s)</b>	<b>Rate:</b> The response data syntax for this parameter is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the current rate of Forward Error Correction (FEC) error. FCCW, selects the FEC-CORR-CW (Forward Error Correction - Correctable - Codeword) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword. FUCW, selects the FEC-UNCORR-CW (Forward Error Correction - Uncorrectable - Codeword) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword. FCSYmb, selects the FEC-CORR-SYMB (Forward Error Correction - Correctable - Symbol) which generates 1 symbol (byte) containing 8 bits in error. FCBIt, selects the FEC-CORR-BIT (Forward Error Correction - Correctable - Bit) which generates 1 symbol (byte) containing 1 bit in error.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE FCCW SOUR:DATA:TEL:OTN:ERR:FEC:AMO 15 SOUR:DATA:TEL:OTN:ERR:FEC:INJ FETC:DATA:TEL:OTN:ERR:FEC:RATE? FCCW
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:FEC:CURRent?

---



**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:TCM[1..n]:HISTory?**

<b>Description</b>	<p>This query returns the history status of the Optical Data Unit (ODU) TCM alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU-TCM</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:TCM[1..n]:HISTory? <wsp>TLTC   TBDI   TTIM   TBIAE   TIAE
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TLTC   TBDI   TTIM   TBIAE   TIAE</p> <p>Sets the history status of the Optical Data Unit (ODU) TCM alarm.</p> <p>TLTC</p> <p>TBDI</p> <p>TTIM</p> <p>TBIAE</p> <p>TIAE</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of Optical Data Unit (ODU) TCM alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:ODU1:TCM1:HIST? TLTC
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:TCM[1..n]:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which the Optical Data Unit (ODU) TCM alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:SEConds? &lt;wsp&gt;TLTC   TBDI   TTIM   TBIAE   TIAE</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TLTC   TBDI   TTIM   TBIAE   TIAE</p> <p>Sets the number of seconds within which the Optical Data Unit (ODU) TCM alarm occurred.</p> <p>TLTC, sets number of seconds of TLTC.</p> <p>TBDI, sets number of seconds of TBDI.</p> <p>TTIM, sets number of seconds of TTIM.</p> <p>TBIAE, sets number of seconds of TBIAE.</p> <p>TIAE, sets number of seconds of TIAE.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of the Optical Data Unit (ODU) TCM alarm.</p> <p>TLTC, sets number of seconds of TLTC.</p> <p>TBDI, sets number of seconds of TBDI.</p> <p>TTIM, sets number of seconds of TTIM.</p> <p>TBIAE, sets number of seconds of TBIAE.</p> <p>TIAE, sets number of seconds of TIAE.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ALAR:ODU1:TCM1:SEC? TLTC</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:CURRent?</p>

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:CURRent?**

<b>Description</b>	<p>This query returns the current status of the Optical Data Unit (ODU) TCM alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:CURRent?          &lt;wsp&gt;TLTC   TBDI   TTIM   TBIAE   TIAE</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TLTC   TBDI   TTIM   TBIAE   TIAE</p> <p>Sets the current status of the Optical Data Unit (ODU) TCM alarm.</p> <p>TLTC          TBDI          TTIM          TBIAE          TIAE</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of the Optical Data Unit (ODU) TCM alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that there is no alarm.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:ODU1:TCM1:CURR? TLTC
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:SEConds?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:HISTory?**

<b>Description</b>	<p>This query returns the history status of Optical Data Unit (ODU) alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)-TCM</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU(1/2)e-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:HISTory? &lt;wsp&gt;TLTC   TBDI   TTIM   TBIAE   TIAE</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TLTC   TBDI   TTIM   TBIAE   TIAE</p> <p>Sets the history status of the Optical Data Unit (ODU) TCM alarm.</p> <p>TLTC</p> <p>TBDI</p> <p>TTIM</p> <p>TBIAE</p> <p>TIAE</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of Optical Data Unit (ODU) alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1:HIST? TLTC</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent?</p>

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which Optical Data Unit (ODU) alarm occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)-TCM</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU(1/2)e-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:SEConds?          &lt;wsp&gt;TLTC   TBDI   TTIM   TBIAE   TIAE</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TLTC   TBDI   TTIM   TBIAE   TIAE</p> <p>Sets the number of seconds within which the Optical Data Unit (ODU) TCM alarm occurred.</p> <p>TLTC, sets number of seconds of TLTC.</p> <p>TBDI, sets number of seconds of TBDI.</p> <p>TTIM, sets number of seconds of TTIM.</p> <p>TBIAE, sets number of seconds of TBIAE.</p> <p>TIAE, sets number of seconds of TIAE.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of Optical Data Unit (ODU) alarm.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1:SEC? TLTC</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:ERRor:[1..n]:E[1..n]:TCM[1..n]:CURRent?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent?**

<b>Description</b>	<p>This query returns the current status of Optical Data Unit (ODU) alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)-TCM</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU(1/2)e-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent? &lt;wsp&gt;TLTC   TBDI   TTIM   TBIAE   TIAE</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TLTC   TBDI   TTIM   TBIAE   TIAE</p> <p>Sets the number of seconds within which the Optical Data Unit (ODU) TCM alarm occurred.</p> <p>TLTC, sets number of seconds of TLTC.</p> <p>TBDI, sets number of seconds of TBDI.</p> <p>TTIM, sets number of seconds of TTIM.</p> <p>TBIAE, sets number of seconds of TBIAE.</p> <p>TIAE, sets number of seconds of TIAE.</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of Optical Data Unit (ODU) alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that there is no alarm.</p> <p>INACTIVE, indicates that the test is not running.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1:CURR? TLTC</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:SECOnds?</p>

<b>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:HISTory?</b>	
<b>Description</b>	<p>This query returns the history status of the Optical Data Unit (ODU) TCM error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU-TCM</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:HISTory? <wsp>TBIP8   TBEI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU TCM error.</p> <p>TBIP8: TBIP8</p> <p>TBEI: TBEI</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of the Optical Data Unit (ODU) TCM error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:ODU1:TCM1:HIST? TBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:OTN:ERRor:ODU[1..n]:TCM[1..n]:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the Optical Data Unit (ODU) TCM error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:OTN:ERRor:ODU[1..n]:TCM[1..n]:SEConds?&lt;wsp&gt;TBIP8   TBEI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU TCM error.</p> <p>TBIP8: TBIP8</p> <p>TBEI: TBEI</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of the Optical Data Unit (ODU) TCM error.</p> <p>TBIP8, selects number of seconds of TBIP8 as the ODU TCM error.</p> <p>TBEI, selects number of seconds of TBEI as the ODU TCM error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ERR:ODU1:TCM1:SEC? TBIP8</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:TCM[1..n]:SEConds?</p>



**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:CURRent?**

<b>Description</b>	<p>This query returns the current status of Optical Data Unit (ODU) TCM error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU-TCM</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:CURRent? <wsp>TBIP8   TBEI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU TCM error.</p> <p>TBIP8: TBIP8</p> <p>TBEI: TBEI</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of the ODU TCM error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:ODU1:TCM1:CURR? TBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:OTN:ERRor:ODU[1..n]:TCM[1..n]:COUNT?

---

<b>Description</b>	<p>This query returns the count of Optical Data Unit (ODU) TCM error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU-TCM</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ERRor:ODU[1..n]:TCM[1..n]:COUNT? <wsp>TBIP8   TBEI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of the ODU TCM error.</p> <p>TBIP8: TBIP8</p> <p>TBEI: TBEI</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of the ODU TCM error.</p> <p>TBIP8, selects TBIP8 as the ODU TCM error.</p> <p>TBEI, selects TBEI as the ODU TCM error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:ODU1:TCM1:COUN? TBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ALARm:ODU[1..n]:TCM[1..n]:SEConds?

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:RATE?**

<b>Description</b>	<p>This query returns the current rate of Optical Data Unit (ODU) TCM error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU-TCM</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:RATE? <wsp>TBIP8   TBEI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU TCM error.</p> <p>TBIP8: TBIP8</p> <p>TBEI: TBEI</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of ODU TCM error.</p> <p>TBIP8, selects TBIP8 as the ODU TCM error.</p> <p>TBEI, selects TBEI as the ODU TCM error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:ODU1:TCM1:RATE? TBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:SEConds?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:HISTory?**

<b>Description</b>	<p>This query returns the history status of Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)-TCM</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU1/2e-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:HISTory? &lt;wsp&gt;TBIP8   TBEI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:HISTory?**

---

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of Optical Data Unit (ODU) error. PRESENT, indicates that at least one error has occurred. ABSENT, indicates that no error occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE TBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ FETC:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:HIST? TBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:COUNT?

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which Optical Data Unit (ODU) error occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)-TCM</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU1/2e-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:SEConds? &lt;wsp&gt;TBIP8   TBEI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE TBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:SEC? TBIP8</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ERRor:[1..n]:E[1..n]:TCM[1..n]:CURRent?</p>

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent?**

---

<b>Description</b>	<p>This query returns the current status of Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)-TCM</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU1/2e-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent? &lt;wsp&gt;TBIP8   TBEI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of Optical Data Unit (ODU) error. PRESENT, indicates that at least one error has occurred in the last second. ABSENT, indicates that there is no error. INACTIVE, indicates that the test is not running.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE TBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ FETC:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:CURR? TBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:CURRent?

---



**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:COUNT?**

<b>Description</b>	<p>This query returns the count of Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)-TCM</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU1/2e-TCM</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:COUNT? <wsp>TBIP8   TBEI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE TBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:COUN? TBIP8</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:[1..n]:E[1..n]:TCM[1..n]:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:RATE?**

<b>Description</b>	<p>This query returns the current rate of Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU3e(1/2)-TCM</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU1/2e-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:RATE? &lt;wsp&gt;TBIP8   TBEI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE TBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ</p> <p>FETC:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:RATE? TBIP8</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:SECOnds?</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:HISTory?**

<b>Description</b>	<p>This query returns the history status of the Generic Mapping Procedure (GMP) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; GMP</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:HISTory? &lt;wsp&gt;GMPOOS[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: GMPOOS</p> <p>Sets the history status of the GMP alarm.</p> <p>GMPOOS: GMPOOS</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the History status of GMP Alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ALAR:HIST? GMPOOS</p> <p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ALAR:HIST? GMPOOS, 3</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU:GMP:RX:ALARm:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:CURRent?**

<b>Description</b>	<p>This query returns the current status of the Generic Mapping Procedure (GMP) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; GMP</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:CURRent? &lt;wsp&gt;GMPOOS[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: GMPOOS</p> <p>Sets the current status of the GMP alarm.</p> <p>GMPOOS: GMPOOS</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Current status of GMP Alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ALAR:CURR? GMPOOS</p> <p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ALAR:CURR? GMPOOS, 3</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ODU:GMP:RX:ALARm:HISTory?</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the Generic Mapping Procedure (GMP) alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; GMP</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ALARm:SEConds? &lt;wsp&gt;GMPOOS[, &lt;Channel&gt;]</pre>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: GMPOOS</p> <p>Sets the seconds value of the GMP alarm.</p> <p>GMPOOS: GMPOOS</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the seconds value of the GMP Alarm.</p>
<b>Example(s)</b>	<pre>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ALAR:SEC? GMPOOS FETC:DATA:TEL:OTN:ODU4:GMP:RX:ALAR:SEC? GMPOOS, 3</pre>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU:GMP:RX:ALARm:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:HISTory?

<b>Description</b>	<p>This query returns the history status of the Generic Mapping Procedure (GMP) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; GMP</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:HISTory? &lt;wsp&gt;CMCRC8   CNDCRC5[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CMCRC8   CNDCRC5</p> <p>Sets the history status of the Generic Mapping Procedure (GMP) error.</p> <p>CMCRC8: CMCRC8</p> <p>CNDCRC5: CNDCRC5</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the History status of GMP Error.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ERR:HIST? CMCRC8</p> <p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ERR:HIST? CMCRC8, 3</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ODU:GMP:RX:ERRor:CURRent?</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:CURRent?**

<b>Description</b>	<p>This query returns the current status of the Generic Mapping Procedure (GMP) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; GMP</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:CURRent?                  &lt;wsp&gt;CMCRC8   CNDCRC5[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CMCRC8   CNDCRC5</p> <p>Sets the current status of the Generic Mapping Procedure (GMP) error.</p> <p>CMCRC8: CMCRC8</p> <p>CNDCRC5: CNDCRC5</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Current status of GMP Error.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ERR:CURR? CMCRC8</p> <p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ERR:CURR? CMCRC8, 3</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ODU:GMP:RX:ERRor:HISTory?</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which the Generic Mapping Procedure (GMP) error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; GMP</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:SEConds? &lt;wsp&gt;CMCRC8   CNDRCRC5[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CMCRC8   CNDRCRC5</p> <p>Sets the number of seconds within which Generic Mapping Procedure (GMP) error occurred.</p> <p>CMCRC8: CMCRC8</p> <p>CNDRCRC5: CNDRCRC5</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the seconds value of the GMP Error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ERR:SEC? CMCRC8</p> <p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ERR:SEC? CMCRC8, 3</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ODU:GMP:RX:ERRor:HISTory?</p>

---



**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:COUNT?**

<b>Description</b>	<p>This query returns the count of the Generic Mapping Procedure (GMP) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; GMP</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:COUNT?          &lt;wsp&gt;CMCRC8   CNDCRC5[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CMCRC8   CNDCRC5</p> <p>Sets the count of the Generic Mapping Procedure (GMP) error.</p> <p>CMCRC8: CMCRC8</p> <p>CNDCRC5: CNDCRC5</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of GMP Error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ERR:COUN? CMCRC8</p> <p>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ERR:COUN? CMCRC8, 3</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU:GMP:RX:ERRor:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:RATE?

---

<b>Description</b>	<p>This query returns the rate value of the Generic Mapping Procedure (GMP) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; GMP</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:ERRor:RATE? &lt;wsp&gt;CMCRC8   CNDRCRC5[, &lt;Channel&gt;]</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CMCRC8   CNDRCRC5</p> <p>Sets the rate value of the Generic Mapping Procedure (GMP) error.</p> <p>CMCRC8: CMCRC8</p> <p>CNDRCRC5: CNDRCRC5</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate value of the GMP Error.</p>
<b>Example(s)</b>	<pre>FETC:DATA:TEL:OTN:ODU4:GMP:RX:ERR:RATE? CMCRC8 FETC:DATA:TEL:OTN:ODU4:GMP:RX:ERR:RATE? CMCRC8, 3</pre>
<b>See Also</b>	<pre>FETCh[1..n]:DATA:TELEcom:OTN:ODU:GMP:RX:ERRor:HISTory?</pre>

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:HISTory?**

<b>Description</b>	<p>This query returns the history status of the Optical Payload Unit (OPU) error.</p> <p>The OPU error occurs only when <code>ìOTU4î</code> is selected as interface/rate and <code>ìODU4(PT=21)/ODU0î</code> or <code>ìODU4(PT=21)/ODU2î</code> are selected as OTN Multiplexing rates.</p> <p>At <code>*RST</code> condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; OPU</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:HISTory? &lt;wsp&gt;OMFI</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a <code>&lt;CHARACTER PROGRAM DATA&gt;</code> element.</p> <p>The allowed elements for this parameter are: OMFI</p> <p>Selects the type of the Optical Payload Unit (OPU) error.</p> <p>OMFI: OPU-OMFI</p>
<b>Response Syntax</b>	<code>&lt;History&gt;</code>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a <code>&lt;CHARACTER RESPONSE DATA&gt;</code> element.</p> <p>Returns the history status of the Optical Payload Unit (OPU) error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:OTN:ERR:OPU1:HIST? OMFI</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CURRent?</code>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which the Optical Payload Unit (OPU) error occurred.</p> <p>The OPU error occurs only when <code>ìOTU4î</code> is selected as interface/rate and <code>ìODU4(PT=21)/ODU0î</code> or <code>ìODU4(PT=21)/ODU2î</code> are selected as OTN Multiplexing rates.</p> <p>At <code>*RST</code> condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; OPU</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:SEConds? <wsp>OMFI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OMFI</p> <p>Selects the type of the Optical Payload Unit (OPU) error.</p> <p>OMFI: OPU-OMFI</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds within which the Optical Payload Unit (OPU) error occurred.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:OPU1:SEC? OMFI
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:HISTory?

---

---

**:FETCh[1..n]:DATA:TELecom:OTN:ERRor:OPU[1..n]:CURRent?**

---

<b>Description</b>	<p>This query returns the current status of the Optical Payload Unit (OPU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; OPU</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ERRor:OPU[1..n]:CURRent? <wsp>OMFI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OMFI</p> <p>Selects the type of the Optical Payload Unit (OPU) error.</p> <p>OMFI: OPU-OMFI</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of the Optical Payload Unit (OPU) error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:OPU1:CURR? OMFI
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ERRor:OPU[1..n]:HISTory?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:OTN:ERRor:OPU[1..n]:COUNT?

---

<b>Description</b>	<p>This query returns the count within which the Optical Payload Unit (OPU) error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; OPU</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ERRor:OPU[1..n]:COUNT? <wsp>OMFI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OMFI</p> <p>Selects the type of the Optical Payload Unit (OPU) error.</p> <p>OMFI: OPU-OMFI</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of the Optical Payload Unit (OPU) error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:OPU1:COUNT? OMFI
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ERRor:OPU[1..n]:HISTory?

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:RATE?**

<b>Description</b>	<p>This query returns the rate at which the Optical Payload Unit (OPU) error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; OPU</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:RATE? <wsp>OMFI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OMFI</p> <p>Selects the type of the Optical Payload Unit (OPU) error.</p> <p>OMFI: OPU-OMFI</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of the Optical Payload Unit (OPU) error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ERR:OPU1:RATE? OMFI
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:HISTory?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:MSIM**

<b>Description</b>	<p>This command enables or disables the Optical Payload Unit-MSIM Monitoring status.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; OPU&gt;MSIM Monitoring</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:MSIM
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables the OPU-PLM (Optical Payload Unit-MSIM Monitoring status).</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OPU1:MSIM ON</p> <p>SENS:DATA:TEL:OTN:OPU1:MSIM?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?

---



**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:MSIM?**

<b>Description</b>	<p>This query returns the status of the Optical Payload Unit-MSIM Monitoring status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path:Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU &gt; OPU&gt;MSIM Monitoring</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:MSIM?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Optical Payload Unit-MSIM Monitoring status.</p> <p>1, Optical Payload Unit-MSIM Monitoring status is enabled.</p> <p>0, Optical Payload Unit-MSIM Monitoring status is disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:OPU1:MSIM ON</p> <p>SENS:DATA:TEL:OTN:OPU1:MSIM?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE**

<b>Description</b>	<p>This command selects the type of Optical Transport Unit (OTU) alarm.</p> <p>At *RST condition, this value is set to OAIS.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE &lt;wsp&gt;OAIS   OBDI   LOF   OOF   LOM   OOM   OBIAE   OIAE   OTIM</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OBDI   LOF   OOF   LOM   OOM   OBIAE   OIAE   OTIM</p> <p>Selects the type of OTU (Optical Transport Unit) alarm.</p> <p>OAIS: OAIS (Optical Transport Unit - Alarm Indication Signal)</p> <p>OBDI: OBDI (Optical Transport Unit - Backward Defect Indication)</p> <p>LOF: LOF (Loss of Frame)</p> <p>OOF: OOF (Out of Frame)</p> <p>LOM: LOM (Loss of Multiframe)</p> <p>OOM: OOM (Out of Multiframe)</p> <p>OBIAE: OTU-BAIE (Optical Transport Unit - Backward Incoming Alignment Error)</p> <p>OIAE: OIAE (Optical Transport Unit - Incoming Alignment Error)</p> <p>OTIM: OTIM</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OTU2:TYPE OAIS</p> <p>SOUR:DATA:TEL:OTN:ALAR:OTU2:TYPE?</p> <p>Returns: OAIS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE?**

<b>Description</b>	<p>This query returns the type of Optical Transport Unit (OTU) alarm.</p> <p>At *RST condition, this value is set to OAIS.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Transport Unit (OTU) alarm.</p> <p>OAIS, Optical Transport Unit - Alarm Indication Signal (OTU-AIS) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OBDI, Optical Transport Unit - Backward Defect Indication (OTU-BDI) is selected as Optical Transport Unit (OTU) alarm.</p> <p>LOF, Loss of Frame (LOF) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OOF, Out of Frame (OOF) is selected as Optical Transport Unit (OTU) alarm.</p> <p>LOM, Loss of Multiframe (LOM) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OOM, Out of Multiframe (OOM) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OBIAE, Optical Transport Unit - Backward Incoming Alignment Error (OTU-BAIE) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OIAE, Optical Transport Unit - Incoming Alignment Error (OTU-ALE) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OTIM, OTIM is selected as Optical Transport Unit (OTU) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OTU2:TYPE OAIS</p> <p>SOUR:DATA:TEL:OTN:ALAR:OTU2:TYPE?</p> <p>Returns: OAIS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]**

<b>Description</b>	<p>This command enables or disables the status of the Optical Transport Unit (OTU) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Inject</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n] &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the OTU (Optical Transport Unit) alarm generation.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OTU2 ON</p> <p>SOUR:DATA:TEL:OTN:ALAR:OTU2?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]?**

<b>Description</b>	<p>This query returns the status of Optical Transport Unit (OTU) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Optical Channel Transport Unit (OTU) alarm generation.</p> <p>1, alarm generation enabled.</p> <p>0, alarm generation disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OTU2 ON</p> <p>SOUR:DATA:TEL:OTN:ALAR:OTU2?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]?</p>

---

## SCPI Command Reference

### *Alarms/Errors*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE**

<b>Description</b>	<p>This command selects the Optical Data Unit (ODU) alarm type for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OAIS.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2))</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE &lt;wsp&gt;OAIS   OBDI   LOF   OOF   LOM   OOM   OBIAe   OIAE</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE**

<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OBDI   LOF   OOF   LOM   OOM   OBIAe   OIAE</p> <p>Selects the Optical Transport Unit (OTU) alarm type.</p> <p>OAIS: Optical Transport Unit - Alarm Indication Signal (OAIS) which generates the polynomial numbers 11 (PN-11) over all OTU frame bits including FAS and MFAS continuously.</p> <p>OBDI: Optical Transport Unit - Backward Defect Indication (OBDI) which generates a ""1"" for the BDI bit in the SM overhead field (byte 3, bit 5) continuously.</p> <p>LOF: Loss of Frame (LOF) which generates the errors in all the FAS bits continuously.</p> <p>OOF: Out of Frame (OOF) which generates the errors in all the FAS bits for 5 consecutive OTU frames repetitively.</p> <p>LOM: Loss of Multiframe (LOM) which generates the errors in multiframe numbers for all the OTU frames continuously.</p> <p>OOM: Out of Multiframe (OOM) which generates the errors in multiframe numbers for 5 consecutive OTU frames repetitively.</p> <p>OBIAe: Optical Transport Unit - Backward Incoming Alignment Error (OTU-BAIE) which generates a ""1011"" for the BEI/BIAE bits in the SM overhead field (byte 3, bits 1 to 4) continuously.</p> <p>OIAE: Optical Transport Unit - Incoming Alignment Error (OIAE) which generates a ""1"" for the IAE bit in the SM overhead field (byte 3, bit 6) continuously.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OTU3:E1:TYPE OAIS</p> <p>SOUR:DATA:TEL:OTN:ALAR:OTU3:E1:TYPE?</p> <p>Returns: OAIS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE?**

<b>Description</b>	<p>This query returns the Optical Data Unit (ODU) alarm type for overlocked rates OTU3e1/2. At *RST condition, this value is set to OAIS.</p> <p>Navigation Path: Test&gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2))</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Optical Transport Unit (OTU) alarm type.</p> <p>OAIS, Optical Transport Unit - Alarm Indication Signal (OTU-AIS) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OBDI, Optical Transport Unit - Backward Defect Indication (OTU-BDI) is selected as Optical Transport Unit (OTU) alarm.</p> <p>LOF2, Loss of Frame (LOF) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OOF1, Out of Frame (OOF) is selected as Optical Transport Unit (OTU) alarm.</p> <p>LOM, Loss of Multiframe (LOM) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OOM, Out of Multiframe (OOM) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OBIAE, Optical Transport Unit - Backward Incoming Alignment Error (OTU-BAIE) is selected as Optical Transport Unit (OTU) alarm.</p> <p>OIAE, Optical Transport Unit - Incoming Alignment Error (OTU-ALE) is selected as Optical Transport Unit (OTU) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OTU3:E1:TYPE OAIS</p> <p>SOUR:DATA:TEL:OTN:ALAR:OTU3:E1:TYPE?</p> <p>Returns: OAIS</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]**

<b>Description</b>	<p>This command enables or disables the Optical Channel Data Unit (ODU) alarm generation for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Alarms) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n] <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Optical Transport Unit (OTU) alarm generation.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ALAR:OTU3:E1:TYPE OAIS SOUR:DATA:TEL:OTN:ALAR:OTU3:E1 ON SOUR:DATA:TEL:OTN:ALAR:OTU3:E1? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]?**

<b>Description</b>	<p>This query returns the status of the Optical Data Unit (ODU) alarm generation for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Alarm) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Optical Transport Unit (OTU) alarm generation.</p> <p>1, alarm generation enabled.</p> <p>0, alarm generation disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OTU3:E1:TYPE OAIS</p> <p>SOUR:DATA:TEL:OTN:ALAR:OTU3:E1 ON</p> <p>SOUR:DATA:TEL:OTN:ALAR:OTU3:E1?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E[1..n]</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE**

<b>Description</b>	<p>This command selects the manual type of Optical Transport Unit (OTU) error.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE <wsp>OBIP8   OBEI   FAS   MFAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the type of OTU error.</p> <p>OBIP8: the type of OTU-BIP-8 (ODU - Bit Interleave Parity-8) error.</p> <p>OBEI: the type of OTU-BEI (ODU - Backward Error Indication) error.</p> <p>FAS: the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS: the type of MFAS (Multiframe Alignment Signal) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU2:MAN:TYPE FAS</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU2:MAN:TYPE?</p> <p>Returns: FAS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type of Optical Transport Unit (OTU) error.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Transport Unit (OTU) error.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (OTU-BIP8) is selected as the Optical Transport Unit (OTU) error.</p> <p>OBEI, ODU - Backward Error Indication (OBE) is selected as the Optical Transport Unit (OTU) error.</p> <p>FAS, Frame Alignment Signal (FAS) is selected as the Optical Transport Unit (OTU) error.</p> <p>MFAS, Multiframe Alignment Signal (MFAS) is selected as the Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU2:MAN:TYPE FAS</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU2:MAN:TYPE?</p> <p>Returns: FAS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOunt**

<b>Description</b>	<p>This command sets the amount of Optical Transport Unit (OTU) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOunt &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of OTU (Optical Transport Unit) error. Choices are 1 to 50. The default setting is 1.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:OTU2:AMO MAX SOUR:DATA:TEL:OTN:ERR:OTU2:AMO?</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOunt SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOunt?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOUnt?**

<b>Description</b>	<p>This query returns the amount of Optical Channel Transport Unit (OTU) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOUnt?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current value of OTU error to inject is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Amount&gt;</p>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU2:AMO MAX</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU2:AMO?</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOUnt</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOUnt?</p>

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:INJect**

---

<b>Description</b>	<p>This command injects the type of Optical Channel Transport Unit (OTU) error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:OTU3:INJ
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:INJect

---

## SCPI Command Reference

### Alarms/Errors

---

<b>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE</b>	
<b>Description</b>	<p>This command selects the manual type Optical Data Unit (ODU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Error) &gt; Defect</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Error) &gt; Defect</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE &lt;wsp&gt;OBIP8   OBEI   FAS   MFAS</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the Optical Transport Unit (OTU) type error.</p> <p>OBIP8: ODU - Bit Interleave Parity-8 (OTU-BIP-8)</p> <p>OBEI: ODU - Backward Error Indication (OTU-BEI)</p> <p>FAS: Frame Alignment Signal (FAS)</p> <p>MFAS: Multiframe Alignment Signal (MFAS)</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE? Returns: OBIP8</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE?</pre>



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type Optical Data Unit (ODU) error for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Error) &gt; Defect</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Error) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Optical Transport Unit (OTU) type error.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (OTU-BIP8) is selected as Optical Transport Unit (OTU) error.</p> <p>OBEI, ODU - Backward Error Indication (OBE) is selected as Optical Transport Unit (OTU) error.</p> <p>FAS, Frame Alignment Signal (FAS) is selected as Optical Transport Unit (OTU) error.</p> <p>MFAS, Multiframe Alignment Signal (MFAS) is selected as Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOut

---

<b>Description</b>	<p>This command sets the amount of Optical Data Unit (ODU) error to be injected for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOut &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of Optical Transport Unit (OTU) error.</p> <p>Choices are 1 through 50.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AMO 15 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AMO? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOut?</pre>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOut?**

<b>Description</b>	<p>This query returns the amount of Optical Channel Data Unit (ODU) error injected for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOut?[\n] <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of error is returned.</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOut</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:INJect**

<b>Description</b>	<p>This command injects the Optical Data Unit (ODU) error type.</p> <p>This command is an event and has no associated *RST condition or query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:INJect
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:INJ</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AMOUnt</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the type of Optical Transport Unit (OTU) error for automated injection.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE <wsp>OBIP8   OBEI   FAS   MFAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS   MFAS</p> <p>Selects the type of Optical Transport Unit (OTU) error for automated injection.</p> <p>OBIP8: the type of OTU-BIP-8 (ODU - Bit Interleave Parity-8) error.</p> <p>OBEI: the type of OTU-BEI (ODU - Backward Error Indication) error.</p> <p>FAS: the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS: the type of MFAS (Multiframe Alignment Signal) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE?</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE?

---

<b>Description</b>	<p>This query returns the type of Optical Transport Unit (OTU) error for automated injection.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Transport Unit (OTU) error for the automated injection.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (OTU-BIP8) is selected as the Optical Transport Unit (OTU) error.</p> <p>OBEI, ODU - Backward Error Indication (OBE) is selected as the Optical Transport Unit (OTU) error.</p> <p>FAS, Frame Alignment Signal (FAS) is selected as the Optical Transport Unit (OTU) error.</p> <p>MFAS, Multiframe Alignment Signal (MFAS) is selected as the Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE**

<b>Description</b>	<p>This command sets the injection rate for the selected Optical Transport Unit (OTU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected Optical Transport Unit (OTU) error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:RATE MIN SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:RATE?</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE?

---

<b>Description</b>	<p>This query returns the injection rate for the selected Optical Transport Unit (OTU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE? [ &lt;wsp&gt;MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current injection rate for the selected OTU error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Rate&gt;</code>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:RATE MIN</code> <code>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:RATE?</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE</code> <code>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE?</code>

---



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated**

<b>Description</b>	<p>This command enables or disables the selected automated Optical Transport Unit (OTU) error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Optical Transport Unit (OTU) error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated?

---

<b>Description</b>	<p>This query returns the status of automated Optical Transport Unit (OTU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated Optical Transport Unit (OTU) error injection.</p> <p>1, OTU error injection is enabled.</p> <p>0, OTU error injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous**

<b>Description</b>	<p>This command enables or disables the continuous rate of automated Optical Transport Unit (OTU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the rate of automated Optical Transport Unit (OTU) error injection as continuous.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:CONT ON SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:CONT? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### **:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous?**

---

<b>Description</b>	<p>This query returns the status of continuous rate of automated Optical Transport Unit (OTU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTU &gt; Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the continuous rate of automated Optical Transport Unit (OTU) error injection.</p> <p>1, OTU continuous error injection enabled.</p> <p>0, OTU continuous error injection disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:CONT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU2:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the Optical Data Unit (ODU) error type for automated injection.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Defect</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE          &lt;wsp&gt;OBIP8   OBEI   FAS1   MFAS</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI   FAS1   MFAS</p> <p>Selects the Optical Transport Unit (OTU) error type for automated injection.</p> <p>OBIP8: the ODU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI: the ODU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS: the Frame Alignment Signal (FAS) error.</p> <p>MFAS: the Multiframe Alignment Signal (MFAS) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the Optical Data Unit (ODU) error type for automated injection.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Defect</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Optical Transport Unit (OTU) type error for automated injection.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (OTU-BIP8) is selected as Optical Transport Unit (OTU) error.</p> <p>OBEI, ODU - Backward Error Indication (OBE) is selected as Optical Transport Unit (OTU) error.</p> <p>FAS1, Frame Alignment Signal (FAS) is selected as Optical Transport Unit (OTU) error.</p> <p>MFAS, Multiframe Alignment Signal (MFAS) is selected as Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE**

---

<b>Description</b>	<p>This command sets the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE <wsp> <Rate>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected Optical Transport Unit (OTU) error.</p> <p>Choices are 1.0E-09 through 6.5E-05.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:RATE?</p> <p>Returns: 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE?</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated</p> <hr/>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE? [ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current injected rate is returned.</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected Optical Transport Unit (OTU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:RATE 1.0E-09 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:RATE? Returns: 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated</p>



**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated**

<b>Description</b>	<p>This command enables or disables the selected automated Optical Data Unit (ODU) error at the rate specified or continuously for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Rate/Maxrate) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Mode (Rate/Maxrate) &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Optical Transport Unit (OTU) error injection.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:RATE 1.0E-09 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT ON SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT? Returns: 1</pre>
<b>See Also</b>	<pre>SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated?</pre>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated?

---

<b>Description</b>	<p>This query returns the status of the automated Optical Data Unit (ODU) error injection for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Rate/Maxrate) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Mode (Rate/Maxrate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated Optical Transport Unit (OTU) error injection.</p> <p>1, OTU error injection is enabled.</p> <p>0, OTU error injection is disabled.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:RATE 1.0E-09 SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT ON SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:RATE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated</pre>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:CONTInuous**

<b>Description</b>	<p>This command enables or disables the automated Optical Data Unit (ODU) error injection rate continuously for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Max Rate</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:CONTInuous <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Optical Transport Unit (OTU) error rate injection continuously.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:CONT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:CONTInuous?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:CONTInuous?**

<b>Description</b>	<p>This query returns the status of the automated Optical Data Unit (ODU) error injection rate for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Max Rate</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e) &gt; Type (Errors) &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the automated Optical Transport Unit (OTU) error rate injection.</p> <p>1, OTU continuous error injection enabled.</p> <p>0, OTU continuous error injection disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:CONT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:OTU3:E1:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E[1..n]:AUTomated:CONTInuous</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:ACHA**

---

<b>Description</b>	<p>This command selects if LO ODU alarm injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:ACHA <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>AllChannel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Selects if LO ODU alarm injection is done on ALL channels, in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU1:COPY
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ALAR:ODU[1..n]:CHAN

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:ACHA?**

<b>Description</b>	<p>This query returns if LO ODU alarm injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:ACHA?
<b>Response Syntax</b>	<AllChannel>
<b>Response(s)</b>	<p><b>AllChannel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns an indication that the LO ODU alarm injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>1, injection is done on all channels.</p> <p>0, injection is done on a specific channel.</p>
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ALAR:ODU[1..n]:CHAN

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CHAN**

---

<b>Description</b>	<p>This command selects the channel used for LO ODU alarm injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CHAN <wsp> <Channel>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel used for LO ODU alarm injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	Multi-Channel OTN Global Copy RX:
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ALAR:ODU[1..n]:ACHA

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CHAN?**

<b>Description</b>	<p>This query returns the channel used for LO ODU alarm injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CHAN?
<b>Response Syntax</b>	<Channel>
<b>Response(s)</b>	<p><b>Channel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the channel used for LO ODU alarm injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>'n', injection is done on channel 'n', when AllCHannel is not selected .</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU100:ACHA ON
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ALAR:ODU[1..n]:ACHA

---



---

**:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE**

---

<b>Description</b>	<p>This command selects the type of Optical Data Unit (ODU) alarm.</p> <p>At *RST condition, this value is set to OAIS.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE &lt;wsp&gt;OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OFSD   OBSD   LOFLom</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

:SOURCE[1..n]:DATA:TELEcom:OTN:ALARM:ODU[1..n]:TYPE

<b>Parameter(s)</b>	<b>Alarm:</b> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OFSD   OBSD   LOFLom</p> <p>Selects the type of ODU (Optical Data Unit) alarm.</p> <p>OAIS: ODU-AIS (ODU - Alarm Indication Signal) which generates an all "1"s pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH), OTUk overhead (OTUk OH) and ODUk FTFL.</p> <p>OBDI: ODU-OCI (ODU - Open Connection Indication) which generates a repeating "01100110" pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH) and OTUk overhead (OTUk OH).</p> <p>OLCK: ODU-LCK (ODU - Locked) which generates a repeating "01010101" pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH) and OTUk overhead (OTUk OH).</p> <p>OOCI: ODU-BDI (ODU - Backward Defect Indication) which generates a "1" in the BDI (byte 3, bit 5) of the PM overhead field continuously.</p> <p>OSFS: ODU-FSF (ODU - Forward Signal Fail) which generates a "00000001" pattern in the FTFL Byte 0 continuously.</p> <p>OBSF: ODU-BSF (ODU - Backward Signal Fail) which generates a "00000001" pattern in the FTFL Byte 128 continuously.</p> <p>OFSD: ODU-FSD (ODU - Forward Signal Degrade) which generates a "00000010" pattern in the FTFL Byte 0 continuously.</p> <p>OBSD: ODU-BSD (ODU - Backward Signal Degrade) which generates a "00000010" pattern in the FTFL Byte 128 continuously.</p> <p>LOFLom, ODU-LOFLom (ODU-Loss of Frame Loss of Multiframe) which generate alarm continuously.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU1:COPY
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:ALARM:OTU[1..n]:TYPE SOURCE[1..n]:DATA:TELEcom:OTN:ALARM:OTU[1..n]:TYPE

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?**

---

<b>Description</b>	<p>This query returns the type of Optical Data Unit (ODU) alarm.</p> <p>At *RST condition, this value is set to OAIS.</p> <p>Navigation Path: Test&gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?**

**Response Syntax** <Alarm>

**Response(s)** **Alarm:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the type of Optical Data Unit (ODU) alarm.

OAIS, ODU-AIS (ODU - Alarm Indication Signal) is selected, which generates an all "1"s pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH), OTUk overhead (OTUk OH) and ODUk FTFL.

OBDI, ODU-OCI (ODU - Open Connection Indication) is selected, which generates a repeating "01100110" pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH) and OTUk overhead (OTUk OH).

OLCK, ODU-LCK (ODU - Locked) is selected, which generates a repeating "01010101" pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH) and OTUk overhead (OTUk OH).

OOCI, ODU-BDI (ODU - Backward Defect Indication) is selected, which generates a "1" in the BDI (byte 3, bit 5) of the PM overhead field continuously.

OFSE, ODU-FSF (ODU - Forward Signal Fail) is selected, which generates a "00000001" pattern in the FTFL Byte 0 continuously.

OBSF, ODU-BSF (ODU - Backward Signal Fail) is selected, which generates a "00000001" pattern in the FTFL Byte 128 continuously.

OFSD, ODU-FSD (ODU - Forward Signal Degrade) is selected, which generates a "00000010" pattern in the FTFL Byte 0 continuously.

OBSD, ODU-BSD (ODU - Backward Signal Degrade) is selected, which generates a "00000010" pattern in the FTFL Byte 128 continuously.

LOFLom, LOFLom is selected.

**See Also**

SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE

SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE?

---

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]**

---

<b>Description</b>	<p>This command enables or disables the status of Optical Channel Data Unit (ODU) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n] <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the ODU (Optical Data Unit) alarm generation.</p>
<b>Example(s)</b>	Multi-Channel OTN Copy RX on channel 2
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]?**

<b>Description</b>	<p>This query returns the status of Optical Data Unit (ODU) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Alarm) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Optical Data Unit (ODU) alarm generation.</p> <p>1, alarm generation enabled.</p> <p>0, alarm generation disabled.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU100:CHAN 2
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]?

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE**

---

**Description**

This command selects the Optical Data Unit (ODU) alarm type for overlocked rates OTU3e1/2.

At \*RST condition, this value is set to OAIS.

Navigation Path: Test > OTN BERT > Results > Alarms/Errors > Global Injection > Layer (ODU3e(1/2))

Navigation Path: Test > OTN BERT > Results > Alarms/Errors > Global Injection > Layer ODU(1/2)e

**Syntax**

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE <wsp>OAIS | OBDI | OLCK | OOCI | OFSF | OBSF | OFSD | OBSD | OTIM | LOFLom

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE**

<b>Parameter(s)</b>	<b>Alarm:</b> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OBDI   OLCK   OOCI   OFSF   OBSF   OFSD   OBSD   OTIM   LOFLom</p> <p>Selects the ODU (Optical Data Unit) alarm type.</p> <p>OAIS: ODU - Alarm Indication Signal (ODU-AIS) which generates an all "1"s pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH), OTUk overhead (OTUk OH) and ODUk FTFL.</p> <p>OBDI: ODU - Open Connection Indication (ODU-OCI) which generates a repeating "01100110" pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH) and OTUk overhead (OTUk OH).</p> <p>OLCK: ODU - Locked which generates a repeating "01010101" pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH) and OTUk overhead (OTUk OH).</p> <p>OOCI: ODU - Backward Defect Indication (ODU-BDI) which generates a "1" in the BDI (byte 3, bit 5) of the PM overhead field continuously.</p> <p>OSFS: ODU - Forward Signal Fail (ODU-FSF) which generates a "00000001" pattern in the FTFL Byte 0 continuously.</p> <p>OBSF: ODU - Backward Signal Fail (ODU-BSF) which generates a "00000001" pattern in the FTFL Byte 128 continuously.</p> <p>OFSD: ODU - Forward Signal Degrade (ODU-FSD) which generates a "00000010" pattern in the FTFL Byte 0 continuously.</p> <p>OBSD: ODU - Backward Signal Degrade (ODU-BSD) which generates a "00000010" pattern in the FTFL Byte 128 continuously.</p> <p>OTIM, ODU-TIM (ODU-TIM) which generates a alarm continuously.</p> <p>LOFLom, ODU-LOFLOM (ODU-Loss of Frame Loss of Multiframe) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU1:COPY
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE?



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE?**

---

<b>Description</b>	<p>This query returns the Optical Data Unit (ODU) alarm type for overclocked rates OTU3e1/2. At *RST condition, this value is set to OAIS.</p> <p>Navigation Path: Test&gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2))</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE?
<b>Response Syntax</b>	<Alarm>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE?**

<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Optical Data Unit (ODU) alarm type.</p> <p>OAIS, ODU - Alarm Indication Signal (ODU-AIS) is selected as Optical Data Unit (ODU) alarm.</p> <p>OBDI, ODU - Backward Defect Indication (ODU-BDI) is selected as Optical Data Unit (ODU) alarm.</p> <p>OLCK, ODU - Locked (ODU-LCK) is selected as Optical Data Unit (ODU) alarm.</p> <p>OOCI, ODU - Open Connection Indication (ODU-OCI) is selected as Optical Data Unit (ODU) alarm.</p> <p>OFSF, ODU - Forward Signal Fail (ODU-FSF) is selected as Optical Data Unit (ODU) alarm.</p> <p>OBSF, ODU - Backward Signal Fail (ODU-BSF) is selected as Optical Data Unit (ODU) alarm.</p> <p>OFSF, ODU - Forward Signal Fail (ODU-FSF) is selected as Optical Data Unit (ODU) alarm.</p> <p>OBSF, ODU - Backward Signal Fail (ODU-BSF) is selected as Optical Data Unit (ODU) alarm.</p> <p>OFSF, ODU - Forward Signal Fail (ODU-FSF) is selected as Optical Data Unit (ODU) alarm.</p> <p>OBSF, ODU - Backward Signal Fail (ODU-BSF) is selected as Optical Data Unit (ODU) alarm.</p> <p>LOFLOM, ODU Loss of Frame Loss of Multiframe (ODU-LOFLOM) is selected as Optical Data Unit (ODU) alarm.</p> <p>LOFLOM, ODU-LOFLOM (ODU-Loss of Frame Loss of Multiframe) is selected as Optical Data Unit (ODU) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TYPE OFSF</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TYPE?</p> <p>Returns: OFSF</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]**

---

<b>Description</b>	<p>This command enables or disables the Optical Channel Data Unit (ODU) alarm generation for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Alarms) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n] <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Optical Data Unit (ODU) alarm generation.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TYPE OFSF SOUR:DATA:TEL:OTN:ALAR:ODU3:E1 ON SOUR:DATA:TEL:OTN:ALAR:ODU3:E1? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]?**

<b>Description</b>	<p>This query returns the status of the Optical Data Unit (ODU) alarm generation for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Alarm) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU2e(1/2)e &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Optical Data Unit (ODU) alarm generation.</p> <p>1, alarm generation enabled.</p> <p>0, alarm generation disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TYPE OFSF</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1 ON</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE**

---

<b>Description</b>	<p>This command selects the manual type of Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Error) &gt; Defect</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE &lt;wsp&gt;OBIP8   OBEI</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU path monitoring sink using the BIP-8 code.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE? Returns: OBIP8</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type of Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Error) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Data Unit (ODU) error.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as the Optical Data Unit (ODU) error.</p> <p>OBEI, ODU - Backward Error Indication (ODU-BEI) is selected as the Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOunt**

---

<b>Description</b>	<p>This command sets the amount of Optical Data Unit (ODU) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOunt &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of ODU (Optical Data Unit) error. Choices are 1 to 50. The default setting is 1.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:ODU1:AMO 15 SOUR:DATA:TEL:OTN:ERR:ODU1:AMO? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOunt SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOunt?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOut?**

<b>Description</b>	<p>This query returns the amount of Optical Channel Data Unit (ODU) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOut?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of Optical Channel Data Unit (ODU) error to inject is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Amount&gt;</p>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOut</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOut?</p>



---

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:INJect**

---

<b>Description</b>	<p>This command injects the type of Optical Data Unit (ODU) error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:INJect
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:INJ</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:INJect

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:ACHA**

<b>Description</b>	<p>This command selects if LO ODU error injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:ACHA <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>AllChannel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Selects if LO ODU error injection is done on ALL channels, in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:ODU1:ACHA ON
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ERR:ODU[1..n]:CHAN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:ACHA?**

<b>Description</b>	<p>This query returns if LO ODU error injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:ACHA?
<b>Response Syntax</b>	<AllChannel>
<b>Response(s)</b>	<p><b>AllChannel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns an indication that the LO ODU error injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>1, injection is done on all channels.</p> <p>0, injection is done on a specific channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:ACHA ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:ACHA?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ERR:ODU[1..n]:CHAN

---

## SCPI Command Reference

### *Alarms/Errors*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CHAN**

<b>Description</b>	<p>This command selects the channel used for LO ODU error injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CHAN &lt;wsp&gt; &lt;Channel&gt;</code>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel used for LO ODU error injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:ERR:ODU1:CHAN 5</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TEL:OTN:ERR:ODU[1..n]:ACHA</code>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CHAN?**

<b>Description</b>	<p>This query returns the channel used for LO ODU error injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:CHAN?
<b>Response Syntax</b>	<Channel>
<b>Response(s)</b>	<p><b>Channel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the channel used for LO ODU error injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>'n', injection is done on channel 'n', when AllCHannel is not selected .</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:CHAN 5</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:CHAN?</p> <p>Returns: 5</p>
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ERR:ODU[1..n]:ACHA

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE**

<b>Description</b>	<p>This command selects the manual type Optical Data Unit (ODU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Error) &gt; Defect</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Error) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE &lt;wsp&gt;OBIP8   OBEI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the manual type Optical Data Unit (ODU) error.</p> <p>OBIP8: the ODU - Bit Interleave Parity-8 (ODU-BIP-8) which indicates the PM BIP-8 mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU - Backward Error Indication (ODU-BEI) which indicates the interleaved block in error detected by the corresponding ODU path monitoring sink using the BIP-8 code.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type Optical Data Unit (ODU) error for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Error) &gt; Defect</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Error) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the manual type Optical Data Unit (ODU) error.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as Optical Data Unit (ODU) error.</p> <p>OBEI, ODU - Backward Error Indication (ODU-BEI) is selected as Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOunt**

<b>Description</b>	<p>This command sets the amount of Optical Data Unit (ODU) error to be injected for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOunt &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element. The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of Optical Data Unit (ODU) error.</p> <p>Choices are 1 through 50.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOunt?</p>



**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOut?**

<b>Description</b>	<p>This query returns the amount of Optical Channel Data Unit (ODU) error injected for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOut?[\n] <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of error is returned.</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOut</pre>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:INJect**

<b>Description</b>	<p>This command injects the Optical Data Unit (ODU) error type.</p> <p>This command is an event and has no associated *RST condition or query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:INJect
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:INJ</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUNT</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the type of Optical Data Unit (ODU) error for automated injection.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE <wsp>OBIP8   OBEI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the type of Optical Data Unit (ODU) error for automated injection.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU path monitoring sink using the BIP-8 code.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the type of Optical Data Unit (ODU) error for automated injection.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Data Unit (ODU) error for the automated injection.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as the Optical Data Unit (ODU) error.</p> <p>OBEI, ODU - Backward Error Indication (ODU-BEI) is selected as the Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE</p> <p>SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE**

<b>Description</b>	<p>This command sets the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE 1.0E-09 SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE? Returns: 1.0E-09</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE? [ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current injection rate for the selected Optical Data Unit (ODU) error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE?</p> <p>Returns: 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated**

<b>Description</b>	<p>This command enables or disables the selected automated Optical Data Unit (ODU) error. At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Mode (Rate/Maxrate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables the automated Optical Data Unit (ODU) error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated?

---

<b>Description</b>	<p>This query returns the status of automated Optical Data Unit (ODU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Mode (Rate/Maxrate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated Optical Data Unit (ODU) error injection.</p> <p>1, ODU error automated injection is enabled.</p> <p>0, ODU error automated injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated?</p>

---



**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous**

<b>Description</b>	<p>This command enables or disables the continuous rate of automated Optical Data Unit (ODU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables the continuous rate of automated Optical Data Unit (ODU) error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:CONT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### **:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous?**

---

<b>Description</b>	<p>This query returns the status of continuous rate of automated Optical Data Unit (ODU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU) &gt; Type (Errors) &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the continuous rate of automated Optical Data Unit (ODU) error injection.</p> <p>1, ODU error continuous mode is enabled.</p> <p>0, ODU error continuous mode is disabled.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:CONT ON SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:CONT? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous?</pre>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the Optical Data Unit (ODU) error type for automated injection.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Defect</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE <wsp>OBIP8   OBEI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OBIP8   OBEI</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8: the ODU - Bit Interleave Parity-8 (ODU-BIP-8) which indicates the PM BIP-8 (Bit-Interleaved Parity - 8 bits) mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU - Backward Error Indication (ODU-BEI) which indicates the interleaved block in error detected by the corresponding ODU (Optical Data Unit) path monitoring sink using the BIP-8 code.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the Optical Data Unit (ODU) error type for automated injection.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Defect</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Optical Data Unit (ODU) error type for automated injection.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as Optical Data Unit (ODU) error.</p> <p>OBEI, ODU - Backward Error Indication (ODU-BEI) is selected as Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TYPE?</p> <p>Returns: OBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE**

<b>Description</b>	<p>This command sets the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:RATE?</p> <p>Returns: 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE?</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the injected rate is returned.</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:RATE 1.0E-09 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:RATE? Returns: 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated**

<b>Description</b>	<p>This command enables or disables the selected automated Optical Data Unit (ODU) error at the rate specified or continuously for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Rate/Maxrate) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Mode (Rate/Maxrate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Optical Data Unit (ODU) error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated?**

<b>Description</b>	<p>This query returns the status of the automated Optical Data Unit (ODU) error injection for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Mode (Rate/Maxrate) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Mode (Rate/Maxrate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the automated Optical Data Unit (ODU) error injection.</p> <p>1, ODU error automated injection is enabled.</p> <p>0, ODU error automated injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p>



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous**

<b>Description</b>	<p>This command enables or disables the automated Optical Data Unit (ODU) error injection rate continuously for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Max Rate</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Optical Data Unit (ODU) error injection rate.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:CONT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous?**

<b>Description</b>	<p>This query returns the status of the automated Optical Data Unit (ODU) error injection rate for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)) &gt; Type (Errors) &gt; Max Rate</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e &gt; Type (Errors) &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the automated Optical Data Unit (ODU) error injection rate.</p> <p>1, ODU error automated injection is enabled.</p> <p>0, ODU error automated injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:CONT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:ACHA**

---

<b>Description</b>	<p>This command selects if LO OPU alarm injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:ACHA <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>AllChannel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Selects if LO OPU alarm injection is done on ALL channels, in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ALAR:OPU1:ACHA ON
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ALAR:OPU[1..n]:CHAN

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:ACHA?**

<b>Description</b>	<p>This query returns if LO OPU alarm injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:ACHA?
<b>Response Syntax</b>	<AllChannel>
<b>Response(s)</b>	<p><b>AllChannel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns an indication that the LO OPU alarm injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>1, injection is done on all channels.</p> <p>0, injection is done on a specific channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OPU1:ACHA ON</p> <p>SOUR:DATA:TEL:OTN:ALAR:OPU1:ACHA?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ALAR:OPU[1..n]:CHAN

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CHAN**

---

<b>Description</b>	<p>This command selects the channel used for LO OPU alarm injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CHAN &lt;wsp&gt; &lt;Channel&gt;</code>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel used for LO OPU alarm injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:ALAR:OPU1:CHAN 5</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TEL:OTN:ALAR:OPU[1..n]:ACHA</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CHAN?**

<b>Description</b>	<p>This query returns the channel used for LO OPU alarm injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:CHAN?
<b>Response Syntax</b>	<Channel>
<b>Response(s)</b>	<p><b>Channel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the channel used for LO OPU alarm injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>'n', injection is done on channel 'n', when AllCHannel is not selected .</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OPU1:CHAN 5</p> <p>SOUR:DATA:TEL:OTN:ALAR:OPU1:CHAN?</p> <p>Returns: 5</p>
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ALAR:OPU[1..n]:ACHA

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE**

<b>Description</b>	<p>This command selects the type of Optical Payload Unit (OPU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE <wsp>OMSIM   OPLM   OAIS   OCSF   OOMFI   LOOMFI
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OMSIM   OPLM   OAIS   OCSF   OOMFI   LOOMFI</p> <p>Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OMSIM: OPU-OMSIM</p> <p>OOMFI: OPU-OOMFI</p> <p>LOOMFI: OPU-LOOMFI</p> <p>OAIS: AIS</p> <p>OCSF: CSF</p> <p>OPLM: PLM</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OAIS</p> <p>SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE?</p> <p>Returns: OAIS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE?**

<b>Description</b>	<p>This query returns the type of Optical Payload Unit (OPU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Payload Unit (OPU) alarm.</p> <p>OMSIM, OPU-OMSIM is selected as the alarm type.</p> <p>OOMFI, OPU-OOMFI is selected as the alarm type.</p> <p>LOOMFI, OPU-LOOMFI is selected as the alarm type.</p> <p>OAIS, OPU-OAIS is selected as the alarm type.</p> <p>OPLM, OPU-OPLM is selected as the alarm type.</p> <p>OCSF, OPU-OCSF is selected as the alarm type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OCSF</p> <p>SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE?</p> <p>Returns: OCSF</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?</p>

---



**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]**

<b>Description</b>	<p>This command enables or disables the status of Optical Payload Unit (OPU) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n] <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the OPU (Optical Payload Unit) alarm generation.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OAIS SOUR:DATA:TEL:OTN:ALAR:OPU1 ON SOUR:DATA:TEL:OTN:ALAR:OPU1? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]?**

<b>Description</b>	<p>This query returns the status of Optical Payload Unit (OPU) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Optical Payload Unit (OPU) alarm generation.</p> <p>1, alarm generation is enabled.</p> <p>0, alarm generation is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OCSF</p> <p>SOUR:DATA:TEL:OTN:ALAR:OPU1 ON</p> <p>SOUR:DATA:TEL:OTN:ALAR:OPU1?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE?</p>

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:MANual:TYPE**

<b>Description</b>	<p>This command selects the manual type of Forward Error Correction (FEC) error.</p> <p>At *RST condition, this value is set to FCCW.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:FEC:MANual:TYPE <wsp>FCCW   FUCW   FCSYmb   FCBit
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCCW   FUCW   FCSYmb   FCBit</p> <p>Selects the type of FEC (Forward Error Correction) error.</p> <p>FCCW: the FEC-CORR-CW (Forward Error Correction - Correctable - Codeword) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW: the FEC-UNCORR-CW (Forward Error Correction - Uncorrectable - Codeword) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.</p> <p>FCSYmb: the FEC-CORR-SYMB (Forward Error Correction - Correctable - Symbol) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit: the FEC-CORR-BIT (Forward Error Correction - Correctable - Bit) which generates 1 symbol (byte) containing 1 bit in error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE?</p> <p>Returns: FCCW</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE</p> <p>SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type Forward Error Correction (FEC) error.</p> <p>At *RST condition, this value is set to FCCW.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Forward Error Correction (FEC) error.</p> <p>FCCW, Forward Error Correction - Correctable - Codeword (FEC-CORR-CW) is selected as the Forward Error Correction (FEC) error.</p> <p>FUCW, Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) is selected as the Forward Error Correction (FEC) error.</p> <p>FCSYMB, Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) is selected as the Forward Error Correction (FEC) error.</p> <p>FCBIT, Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) is selected as the Forward Error Correction (FEC) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE?</p> <p>Returns: FCCW</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AMOUnt**

<b>Description</b>	<p>This command sets the amount of Forward Error Correction (FEC) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AMOUnt &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of FEC (Forward Error Correction) error. Choices are 1 to 50. The default setting is 1.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE FCCW SOUR:DATA:TEL:OTN:ERR:FEC:AMO 15 SOUR:DATA:TEL:OTN:ERR:FEC:AMO? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOUnt SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOUnt?</pre>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AMOUt?

---

<b>Description</b>	<p>This query returns the amount of Forward Error Correction (FEC) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AMOUt?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of Forward Error Correction (FEC) error to inject is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of Forward Error Correction (FEC) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOUt</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOUt?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:INJect**

---

<b>Description</b>	<p>This command injects the type of Forward Error Correction (FEC) error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:INJect
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:MAN:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:INJ</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:INJect

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the Forward Error Correction (FEC) error type for automated injection.</p> <p>At *RST condition, this value is set to FCCW.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:TYPE &lt;wsp&gt;FCCW   FUCW   FCSYmb   FCBit</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCCW   FUCW   FCSYmb   FCBit</p> <p>Selects the Forward Error Correction (FEC) error type for automated injection.</p> <p>FCCW: the Forward Error Correction - Correctable - Codeword (FEC-CORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW: the Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.</p> <p>FCSYmb: the Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit: the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1 bit in error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:TYPE?</p> <p>Returns: FCCW</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE?</p>



**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the Forward Error Correction (FEC) error type for automated injection.</p> <p>At *RST condition, this value is set to FCCW.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Forward Error Correction (FEC) error type for the automated injection.</p> <p>FCCW, Forward Error Correction - Correctable - Codeword (FEC-CORR-CW) is selected.</p> <p>FUCW, Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) is selected.</p> <p>FCSYmb, Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) is selected.</p> <p>FCBit, Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:TYPE?</p> <p>Returns: FCCW</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:RATE**

<b>Description</b>	<p>This command sets the injection rate for the selected Forward Error Correction (FEC) error. At *RST condition, this value is set to 1.5E-02.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element. The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected Forward Error Correction (FEC) error. Choices are 1.0E-07 to 1.5E-02.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:RATE?</p> <p>Returns: 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the injection rate for the selected Forward Error Correction (FEC) error. At *RST condition, this value is set to 1.5E-02.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Mode (Manual) &gt; Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:RATE?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current injection rate for the selected Forward Error Correction (FEC) error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected Forward Error Correction (FEC) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:RATE?</p> <p>Returns: 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE?</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated

---

<b>Description</b>	<p>This command enables or disables the selected automated Forward Error Correction (FEC) error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Forward Error Correction (FEC) error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated?**

<b>Description</b>	<p>This query returns the status of the automated Forward Error Correction (FEC) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the automated Forward Error Correction (FEC) error injection.</p> <p>1, FEC error injection is enabled.</p> <p>0, FEC error injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:CONTInuous

<b>Description</b>	<p>This command enables or disables the rate of automated Forward Error Correction (FEC) error injection continuously.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Mode (MaxRate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:CONTInuous <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Forward Error Correction (FEC) error rate injection continuously.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:TYPE FCCW SOUR:DATA:TEL:OTN:ERR:FEC:AUT:CONT ON SOUR:DATA:TEL:OTN:ERR:FEC:AUT:CONT?</pre> <p>Returns: 1</p>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous?</pre>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:CONTInuous?**

---

<b>Description</b>	<p>This query returns the status of the automated Forward Error Correction (FEC) error rate injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (FEC) &gt; Type (Errors) &gt; Mode (MaxRate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:FEC:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the automated Forward Error Correction (FEC) error rate injection.</p> <p>1, FEC continuous error injection is enabled.</p> <p>0, FEC continuous error injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:TYPE FCCW</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:CONT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:FEC:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:MANual:TYPE**

<b>Description</b>	<p>This command selects the manual type of Optical Payload Unit (OPU) error.</p> <p>At *RST condition, this value is set to OMFI.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:MANual:TYPE &lt;wsp&gt;OMFI</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OMFI</p> <p>Selects the type of Optical Payload Unit (OPU) error.</p> <p>OMFI</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OPU1:MAN:TYPE OMFI</p> <p>SOUR:DATA:TEL:OTN:ERR:OPU1:MAN:TYPE?</p> <p>Returns: OMFI</p>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE</code>

---



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:MANual:TYPE?**

---

<b>Description</b>	<p>This query returns the manual type of Optical Payload Unit (OPU) error.</p> <p>At *RST condition, this value is set to OMFI.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Payload Unit (OPU) error.</p> <p>OMFI, OMFI is selected as the Optical Payload Unit Error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OPU1:MAN:TYPE OMFI</p> <p>SOUR:DATA:TEL:OTN:ERR:OPU1:MAN:TYPE?</p> <p>Returns: OMFI</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOunt**

<b>Description</b>	<p>This command sets the amount of Optical Payload Unit (OPU) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOunt &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of Optical Payload Unit (OPU) error. Choices are 1 to 50.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OPU4:AMO 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOunt</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOUnt?**

---

<b>Description</b>	<p>This query returns the amount of Optical Payload Unit (OPU) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AMOUnt?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of OPU error to inject is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of Optical Payload Unit (OPU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OPU4:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOUnt?

---

## SCPI Command Reference

### *Alarms/Errors*

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:INJect**

---

<b>Description</b>	<p>This command injects the type of Optical Payload Unit (OPU) error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:OPU4:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:INJect

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:ACHA**

<b>Description</b>	<p>This command selects if LO OPU error injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:ACHA <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>AllChannel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Selects if LO OPU error injection is done on ALL channels, in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:OPU1:ACHA ON
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ERR:OPU[1..n]:CHAN

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:ACHA?**

<b>Description</b>	<p>This query returns if LO OPU error injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:ACHA?
<b>Response Syntax</b>	<AllChannel>
<b>Response(s)</b>	<p><b>AllChannel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns an indication that the LO OPU error injection is done on ALL channels, in the Multi-Channel OTN test application.</p> <p>1, injection is done on all channels.</p> <p>0, injection is done on a specific channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OPU1:ACHA ON</p> <p>SOUR:DATA:TEL:OTN:ERR:OPU1:ACHA?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ERR:OPU[1..n]:CHAN

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CHAN**

---

<b>Description</b>	<p>This command selects the channel used for LO OPU error injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CHAN &lt;wsp&gt; &lt;Channel&gt;</code>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel used for LO OPU error injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:ERR:OPU1:CHAN 5</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TEL:OTN:ERR:OPU[1..n]:ACHA</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CHAN?**

<b>Description</b>	<p>This query returns the channel used for LO OPU error injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:CHAN?
<b>Response Syntax</b>	<Channel>
<b>Response(s)</b>	<p><b>Channel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the channel used for LO OPU error injection, when AllCHannel is not selected, in the Multi-Channel OTN test application.</p> <p>'n', injection is done on channel 'n', when AllCHannel is not selected .</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OPU1:CHAN 5</p> <p>SOUR:DATA:TEL:OTN:ERR:OPU1:CHAN?</p> <p>Returns: 5</p>
<b>See Also</b>	SOURce[1..n]:DATA:TEL:OTN:ERR:OPU[1..n]:ACHA

---



**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the type of Optical Payload Unit (OPU) error for automated injection. At *RST condition, this value is set to OMFI.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE &lt;wsp&gt;OMFI</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OMFI</p> <p>Selects the type of Optical Payload Unit (OPU) error for automated injection.</p> <p>OMFI</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:OPU4:AUT:TYPE OMFI SOUR:DATA:TEL:OTN:ERR:OPU4:AUT:TYPE? Returns: OMFI</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the type of Optical Payload Unit (OPU) error for automated injection.</p> <p>At *RST condition, this value is set to OMFI.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Payload Unit (OPU) error for the automated injection.</p> <p>OMFI is selected as the type of OPU error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OPU4:AUT:TYPE OMFI</p> <p>SOUR:DATA:TEL:OTN:ERR:OPU4:AUT:TYPE?</p> <p>Returns: OMFI</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE**

<b>Description</b>	<p>This command sets the injection rate for the selected Optical Payload Unit (OPU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; MODE (Rate) &gt; Rate</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected Optical Payload Unit (OPU) error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:ERR:OPU4:AUT:RATE 1.0E-09</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the injection rate for the selected Optical Payload Unit (OPU) error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; MODE (Rate) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:RATE?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current injection rate for the selected OPU error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected Optical Payload Unit (OPU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OPU4:AUT:RATE 1.0E-09</p> <p>SOUR:DATA:TEL:OTN:ERR:OPU4:AUT:RATE?</p> <p>Returns: 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:RATE?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated**

<b>Description</b>	<p>This command enables or disables the selected automated Optical Payload Unit (OPU) error. At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Optical Payload Unit (OPU) error injection.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:OPU4:AUT ON
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated?

---

<b>Description</b>	<p>This query returns the status of automated Optical Payload Unit (OPU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated Optical Payload Unit (OPU) error injection.</p> <p>1, automated mode is enabled.</p> <p>0, automated mode is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OPU4:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated?

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous**

<b>Description</b>	<p>This command enables or disables the continuously rate of automated Optical Payload Unit (OPU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Mode (MaxRate)</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the continuous rate of automated Optical Payload Unit (OPU) error injection.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:OPU4:AUT:CONT ON</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### **:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous?**

---

<b>Description</b>	<p>This query returns the status of continuously rate of automated Optical Payload Unit (OPU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU) &gt; Type (Errors) &gt; Mode (MaxRate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OPU[1..n]:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of continuously rate of automated Optical Payload Unit (OPU) error injection.</p> <p>1, OPU continuous mode is enabled.</p> <p>0, OPU continuous mode is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:OPU4:AUT:CONT 1</p> <p>SOUR:DATA:TEL:OTN:ERR:OPU4:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AUTomated:CONTInuous?

---



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE**

---

<b>Description</b>	<p>This command selects the type of Optical Data Unit (ODU) TCM alarm.</p> <p>At *RST condition, this value is set to TLTC.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE <wsp>TLTC   TBDI   TIAE   TBIAE
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TLTC   TBDI   TIAE   TBIAE</p> <p>Selects the type of ODU (Optical Data Unit) TCM alarm.</p> <p>TLTC</p> <p>TBDI</p> <p>TIAE</p> <p>TBIAE</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ALAR:ODU3:TCM1:TYPE TLTC
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE?	
<b>Description</b>	<p>This query returns the type of Optical Data Unit (ODU) TCM alarm.</p> <p>At *RST condition, this value is set to TLTC.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the ODU TCM alarm type value.</p> <p>TLTC, selects the type of ODU TCM alarm as TLTC.</p> <p>TBDI, selects the type of ODU TCM alarm as TBDI.</p> <p>TIAE, selects the type of ODU TCM alarm as TIAE.</p> <p>TBIAE, selects the type of ODU TCM alarm as TBIAE.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:ODU3:TCM1:TYPE TLTC</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU3:TCM1:TYPE?</p> <p>Returns: TLTC</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]**

---

<b>Description</b>	<p>This command enables or disables the status of Optical Channel Data Unit (ODU) TCM alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n] <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Level:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the ODU (Optical Data Unit) TCM alarm generation.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:ODU4:TCM1 ON</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU4:TCM1?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]?**

<b>Description</b>	<p>This query returns the status of Optical Data Unit (ODU) TCM alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Optical Data Unit (ODU) TCM alarm generation.</p> <p>1, enables the ODU TCM alarm generation.</p> <p>0, disables the ODU TCM alarm generation.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ALAR:ODU4:TCM1?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]?

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:TYPE**

<b>Description</b>	<p>This command selects the type of Optical Data Unit (ODU) alarm.</p> <p>At *RST condition, this value is set to OAIS.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:TYPE &lt;wsp&gt;TLTC   TBDI   TIAE   TBIAE</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TLTC   TBDI   TIAE   TBIAE</p> <p>Selects the type of ODU (Optical Data Unit) TCM alarm.</p> <p>TLTC</p> <p>TBDI</p> <p>TIAE</p> <p>TBIAE</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1:TYPE TLTC</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1:TYPE?</p> <p>Returns: TLTC</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:TYPE?**

<b>Description</b>	<p>This query returns the type of Optical Data Unit (ODU) alarm.</p> <p>At *RST condition, this value is set to OAIS.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Data Unit (ODU) alarm.</p> <p>OAIS, ODU - Alarm Indication Signal (ODU-AIS) is selected as Optical Data Unit (ODU) alarm.</p> <p>OBDI, ODU - Backward Defect Indication (ODU-BDI) is selected as Optical Data Unit (ODU) alarm.</p> <p>OLCK, ODU - Locked (ODU-LCK) is selected as Optical Data Unit (ODU) alarm.</p> <p>OOCI, ODU - Open Connection Indication (ODU-OCI) is selected as Optical Data Unit (ODU) alarm.</p> <p>OFSF, ODU - Forward Signal Fail (ODU-FSF) is selected as Optical Data Unit (ODU) alarm.</p> <p>OBSF, ODU - Backward Signal Fail (ODU-BSF) is selected as Optical Data Unit (ODU) alarm.</p> <p>OFSD, ODU - Forward Signal Degrade (ODU-FSD) is selected as Optical Data Unit (ODU) alarm.</p> <p>OBSD, ODU - Backward Signal Degrade (ODU-BSD) is selected as Optical Data Unit (ODU) alarm.</p> <p>LOFLOM, ODU Loss of Frame Loss of Multiframe (ODU-LOFLOM) is selected as Optical Data Unit (ODU) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1:TYPE TLTC</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1:TYPE?</p> <p>Returns: TLTC</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]**

<b>Description</b>	<p>This command enables or disables the status of Optical Channel Data Unit (ODU) TCM alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n] <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the ODU (Optical Data Unit) TCM alarm generation.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1:TYPE OFSF</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1 ON</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]?**

<b>Description</b>	<p>This query returns the status of Optical Data Unit (ODU) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TCM[1..n]?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Optical Data Unit (ODU) alarm generation.</p> <p>1, enables the ODU TCM alarm generation.</p> <p>0, disables the ODU TCM alarm generation.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1:TYPE OFSF</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1 ON</p> <p>SOUR:DATA:TEL:OTN:ALAR:ODU3:E1:TCM1?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E[1..n]</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE**

<b>Description</b>	<p>This command selects the manual type of Optical Data Unit (ODU) TCM error.</p> <p>At *RST condition, this value is set to TBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE &lt;wsp&gt;TBIP8   TBEI</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU TCM error.</p> <p>TBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 mismatch between the received value and the locally computed value (0 to 8).</p> <p>TBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU path monitoring sink using the BIP-8 code.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU3:MAN:TCM1:TYPE TBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:MAN:TCM1:TYPE? Returns: TBIP8</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE?**

<b>Description</b>	<p>This query returns the manual type of Optical Data Unit (ODU) TCM error.</p> <p>At *RST condition, this value is set to TBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Data Unit (ODU) TCM error.</p> <p>TBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as the Optical Data Unit (ODU)-TCM error.</p> <p>TBEI, ODU - Backward Error Indication (ODU-BEI) is selected as the Optical Data Unit (ODU)-TCM error.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:ODU3:MAN:TCM1:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOut**

<b>Description</b>	<p>This command sets the amount of ODU-TCM error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOut &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of ODU (Optical Data Unit) TCM error. Choices are 1 to 50.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU3:TCM1:AMO 1 SOUR:DATA:TEL:OTN:ERR:ODU3:TCM1:AMO? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOut</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOut?**

<b>Description</b>	<p>This query returns the amount of ODU-TCM error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOut? [ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of ODU-TCM error to inject is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Amount&gt;</p>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of Optical Data Unit (ODU) TCM error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:TCM1:AMO?</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOut?</p>

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:INJect**

---

<b>Description</b>	<p>This command injects the type of ODU-TCM error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:ODU3:TCM1:INJ
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:INJect

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TCM[1..n]:TYPE**

<b>Description</b>	<p>This command selects the manual type of Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)-TCM) &gt; Type (Errors) &gt; Defect</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e-TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TCM[1..n]:TYPE &lt;wsp&gt;TBIP8   TBEI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of ODU (Optical Data Unit) error.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU path monitoring sink using the BIP-8 code.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TCM1:TYPE TBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TCM1:TYPE?</p> <p>Returns: TBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE?</p>

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TCM[1..n]:TYPE?**

---

<b>Description</b>	<p>This query returns the manual type of Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is set to OBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU3e(1/2)-TCM) &gt; Type (Errors) &gt; Defect</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU(1/2)e-TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TCM[1..n]:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Data Unit (ODU) error.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as Optical Data Unit (ODU) error.</p> <p>OBEI, ODU - Backward Error Indication (ODU-BEI) is selected as Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TCM1:TYPE TBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TCM1:TYPE?</p> <p>Returns: TBIP8</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AMOunt**

<b>Description</b>	<p>This command sets the amount of Optical Data Unit (ODU) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU3e(1/2)-TCM &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e-TCM &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AMOunt &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of ODU (Optical Data Unit) error.</p> <p>Choices are 1 through 50. The default setting is 1.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TCM1:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOunt?</p>



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AMOUnt?**

<b>Description</b>	<p>This query returns the amount of Optical Channel Data Unit (ODU) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU3e(1/2)-TCM &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e-TCM &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AMOUnt?{<wsp>MAXimum   MINimum}
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of error is returned.</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TCM1:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUnt</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:INJect**

<b>Description</b>	<p>This command injects the type of Optical Data Unit (ODU) error.</p> <p>This command is an event and has no associated *RST condition or query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU3e(1/2)-TCM &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e-TCM &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:INJect
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:MAN:TCM1:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:AMO 15</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:INJ</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AMOUnt</p>

---

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:TYPE**

---

<b>Description</b>	<p>This command selects the type of ODU-TCM error for automated injection.</p> <p>At *RST condition, this value is set to TBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:TYPE &lt;wsp&gt;TBIP8   TBEI</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of Optical Data Unit (ODU) TCM error for automated injection.</p> <p>TBIP8: the ODU TCM-BIP-8 (ODU TCM - Bit Interleave Parity-8) which indicates the PM BIP-8 mismatch between the received value and the locally computed value (0 to 8).</p> <p>TBEI: the ODU TCM-BEI (ODU TCM - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU path monitoring sink using the BIP- 8 code.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU3:AUT:TCM1:TYPE TBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:AUT:TCM1:TYPE?</pre> <p>Returns: TBIP8</p>
<b>See Also</b>	<pre>SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:TYPE?	
<b>Description</b>	<p>This query returns the type of ODU-TCM error for automated injection.</p> <p>At *RST condition, this value is set to TBIP8.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Data Unit (ODU) TCM error for the automated injection.</p> <p>TBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as the Optical Data Unit (ODU)-TCM error.</p> <p>TBEI, ODU - Backward Error Indication (ODU-BEI) is selected as the Optical Data Unit (ODU)-TCM error.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:ODU3:AUT:TCM1:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE?

---

```
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:TYPE
```

---

<b>Description</b>	<p>This command selects the type of Optical Data Unit (ODU) error for automated injection for overclocked rates.</p> <p>At *RST condition, this value is set to BERRor.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(3e1/3e2/1e/2e)-TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:TYPE <wsp>TBIP8   TBEI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TBIP8   TBEI</p> <p>Selects the type of Optical Data Unit (ODU) error for automated injection.</p> <p>OBIP8: the ODU-BIP-8 (ODU - Bit Interleave Parity-8) which indicates the PM BIP-8 mismatch between the received value and locally computed value (0 to 8).</p> <p>OBEI: the ODU-BEI (ODU - Backward Error Indication) which indicates the interleaved block in error detected by the corresponding ODU path monitoring sink using the BIP-8 code.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:TYPE TBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:TYPE?</pre> <p>Returns: TBIP8</p>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE? SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:TYPE?**

<b>Description</b>	<p>This query returns the type of Optical Data Unit (ODU) error for automated injection.</p> <p>At *RST condition, this value is set to BERRor.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(3e1/3e2/1e/2e)-TCM &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:TYPE?</code>
<b>Response Syntax</b>	<code>&lt;Error&gt;</code>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Data Unit (ODU) TCM error for the automated injection.</p> <p>TBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as the Optical Data Unit (ODUe1/2)-TCM error.</p> <p>TBEI, ODU - Backward Error Indication (ODU-BEI) is selected as the Optical Data Unit (ODUe1/2)-TCM error.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:TYPE TBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:TYPE? Returns: TBIP8</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated</pre>

---

---

```
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:RATE
```

---

<b>Description</b>	<p>This command sets the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(3e1/3e2/1e/2e)-TCM &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:RATE <wsp> <Rate>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected Optical Data Unit (ODU) error.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:RATE 1.0E-09 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:RATE? Returns: 1.0E-09</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE? SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:RATE?**

<b>Description</b>	<p>This query returns the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(3e1/3e2/1e/2e)-TCM &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:RATE?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the injected rate is returned.</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected Optical Data Unit (ODU) TCM error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:TYPE OBIP8 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:RATE 1.0E-09 SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:RATE? Returns: 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p>



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:RATE**

---

<b>Description</b>	<p>This command sets the injection rate for the selected ODU-TCM error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected Optical Data Unit (ODU) TCM error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU3:AUT:TCM1:RATE 1.0E-09</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:RATE?	
<b>Description</b>	<p>This query returns the injection rate for the selected ODU-TCM error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:RATE? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current injection rate for the selected ODU-TCM error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected Optical Data Unit (ODU) TCM error.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:ODU3:AUT:TCM1:RATE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:RATE?

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AUTomated**

---

<b>Description</b>	<p>This command enables or disables the selected automated ODU-TCM error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU-TCM &gt; Type (Errors) &gt; Mode (Rate/Max Rate) &gt; Inject</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AUTomated &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Optical Data Unit (ODU) TCM error injection.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ERR:ODU3:TCM1:AUT ON</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AUTomated?**

<b>Description</b>	<p>This query returns the status of automated ODU-TCM error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU-TCM &gt; Type (Errors) &gt; Mode (Rate/Max Rate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated Optical Data Unit (ODU) TCM alarm generation.</p> <p>1, ODU TCM error injection is enabled.</p> <p>0, ODU TCM error injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:TCM1:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:TCM1:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated?

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AUTomated**

<b>Description</b>	<p>This command enables or disables the selected automated Optical Data Unit (ODU) error at the rate specified or continuously.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU3e(1/2)-TCM &gt; Type (Errors) &gt; Mode (Rate/Max Rate) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e-TCM &gt; Type (Errors) &gt; Mode (Rate/Max Rate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Optical Data Unit (ODU) error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AUTomated?**

<b>Description</b>	<p>This query returns the status of automated Optical Data Unit (ODU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU3e(1/2)-TCM &gt; Type (Errors) &gt; Mode (Rate/Max Rate) &gt; Inject</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e-TCM &gt; Type (Errors) &gt; Mode (Rate/Max Rate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:TCM[1..n]:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated Optical Data Unit (ODU) error injection.</p> <p>1, ODU TCM error injection is enabled.</p> <p>0, ODU TCM error injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:AUT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:TCM1:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous**

<b>Description</b>	<p>This command enables or disables the continuously rate of automated ODU-TCM error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the continuous rate of automated Optical Data Unit (ODU) TCM error injection.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:ERR:ODU3:AUT:TCM1:CONT ON</code>
<b>See Also</b>	<code>SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous?**

<b>Description</b>	<p>This query returns the status of the continuous rate of automated ODU-TCM error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (ODU-TCM) &gt; Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:CONTInous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the continuous rate of automated Optical Data Unit (ODU) TCM error injection.</p> <p>1, ODU TCM continuous mode is enabled.</p> <p>0, ODU TCM continuous mode is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:ERR:ODU3:AUT:TCM1:CONT?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:CONTInuous?

---



---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:CONTInuous**


---

<b>Description</b>	<p>This command enables or disables the continuously rate of automated Optical Data Unit (ODU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU3e(1/2)-TCM &gt; Type (Errors) &gt; Mode (Max Rate)</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e-TCM &gt; Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:CONTInuous <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the continuous rate of automated Optical Data Unit (ODU) error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:CONT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous?</p> <hr/>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:CONTInuous?**

<b>Description</b>	<p>This query returns the status of continuously rate of automated Optical Data Unit (ODU) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU3e(1/2)-TCM &gt; Type (Errors) &gt; Mode (Max Rate)</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer ODU(1/2)e-TCM &gt; Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TCM[1..n]:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of continuously rate of automated Optical Data Unit (ODU) error injection.</p> <p>1, ODU TCM continuous mode is enabled.</p> <p>0, ODU TCM continuous mode is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:TYPE OBIP8</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:CONT ON</p> <p>SOUR:DATA:TEL:OTN:ERR:ODU3:E1:AUT:TCM1:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E[1..n]:AUTomated:CONTInuous</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm:TYPE**

<b>Description</b>	<p>This command selects the type of OTL global alarm.</p> <p>At *RST condition, this value is set to LOL.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm:TYPE <wsp>LOL
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOL</p> <p>Selects the type of OTL global alarm.</p> <p>LOL</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:GLOB:ALAR:TYPE LOL
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm:TYPE?**

<b>Description</b>	<p>This query returns the type of OTL global alarm.</p> <p>At *RST condition, this value is set to LOL.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of OTL global alarm.</p> <p>LOL, LOL is selected as the OTL alarm.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:GLOB:ALAR:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE?

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm**

<b>Description</b>	<p>This command enables or disables the OTL global alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Level:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the OTL global alarm generation.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:GLOB:ALAR ON
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n] SOURCE[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm?**

<b>Description</b>	<p>This query returns the enabled or disabled OTL global alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of OTL global alarm generation.</p> <p>1, global alarm generation is enabled.</p> <p>0, global alarm generation is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:GLOB:ALAR?
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]?</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALARm:TYPE**

<b>Description</b>	<p>This command selects the type of OTL alarm.</p> <p>At *RST condition, this value is set to LOL.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALARm:TYPE &lt;wsp&gt;LOF   LOR   OOF   OOR</code>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF   LOR   OOF   OOR</p> <p>Selects the type of OTL alarm.</p> <p>LOF</p> <p>LOR</p> <p>OOF</p> <p>OOR</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:OTL:ALAR:TYPE LOF</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm:TYPE</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALARm:TYPE?**

<b>Description</b>	<p>This query returns the type of OTL alarm.</p> <p>At *RST condition, this value is set to LOL.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALARm:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of OTL alarm.</p> <p>LOF, LOF is selected as the type of OTL alarm.</p> <p>LOR, LOR is selected as the type of OTL alarm.</p> <p>OOF, OOF is selected as the type of OTL alarm.</p> <p>OOR, OOR is selected as the type of OTL alarm.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:ALAR:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm:TYPE?

---



---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALARm**

---

<b>Description</b>	<p>This command enables or disables the OTL alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALARm &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Level:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the OTL alarm generation.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:OTL:ALAR ON</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALARm?**

<b>Description</b>	<p>This query returns the enabled or disabled OTL alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALARm?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of OTL alarm generation.</p> <p>1, alarm generation is enabled.</p> <p>0, alarm generation is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:ALAR?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTL:GLOBal:ALARm?

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:MANual:TYPE**

<b>Description</b>	<p>This command selects the manual type of OTL error.</p> <p>At *RST condition, this value is set to FAS error.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:MANual:TYPE <wsp>FAS   LLM
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the type of OTL manual error.</p> <p>FAS</p> <p>LLM</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:ERR:MAN:TYPE FAS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type of OTL error.</p> <p>At *RST condition, this value is set to FAS error.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of manual OTL error.</p> <p>FAS, FAS is selected as the manual error type.</p> <p>LLM, LLM is selected as the manual error type.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:ERR:MAN:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AMOUNT**

<b>Description</b>	<p>This command sets the amount for the OTL error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AMOUNT &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the OTL error amount value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:OTL:ERR:AMO 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOUNT</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AMOut?**

<b>Description</b>	<p>This query returns the amount value for the OTL error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AMOut? &lt;wsp&gt;MAXimum   MINimum</code>
<b>Response Syntax</b>	<code>&lt;Amount&gt;</code>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of OTL error.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:OTL:ERR:AMO?</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOut?</code>

---

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:INJect**

---

<b>Description</b>	<p>This command injects the OTL error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:ERRor:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:ERR:INJ
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:ODU:INJect

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the automated type of OTL error.</p> <p>At *RST condition, this value is set to FAS.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:TYPE &lt;wsp&gt;FAS   LLM</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FAS   LLM</p> <p>Selects the type of OTL automated error.</p> <p>FAS</p> <p>LLM</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:OTL:ERR:AUT:TYPE FAS</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:TYPE</code>



**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the automated type of OTL error.</p> <p>At *RST condition, this value is set to FAS.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of automated OTL error.</p> <p>FAS, FAS is selected as the type of automated error.</p> <p>LLM, LLM is selected as the type of automated error.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:ERR:AUT:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:RATE**

<b>Description</b>	<p>This command sets the automated rate for the OTL error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the OTL error rate value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTL:ERR:AUT:RATE 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:RATE</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the automated rate value for the OTL error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:RATE? <wsp>MAXimum   MINimum
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of automated OTL error.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:ERR:AUT:RATE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:RATE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:CONTInuous**

<b>Description</b>	<p>This command sets the continuous mode status of the automated error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Mode (Maxrate)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the OTL continuous mode status value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTL:ERR:AUT:CONT ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:CONTInuous?**

<b>Description</b>	<p>This query returns the continuous mode status of the automated error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Mode (Maxrate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated error continuous mode.</p> <p>1, continuous mode is enabled.</p> <p>0, continuous mode is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:ERR:AUT:CONT?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated**

<b>Description</b>	<p>This command sets the automated error generation status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Mode(Maxrate/Rate)/Inject</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the OTL error generation status value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTL:ERR:AUT ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated?**

<b>Description</b>	<p>This query returns the automated error generation status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OTL) &gt; Type (Errors) &gt; Mode(Maxrate/Rate)/Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ERRor:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of OTL automated error generation.</p> <p>1, error generation is enabled.</p> <p>0, error generation is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:ERR:AUT?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AUTomated?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:LANE**

<b>Description</b>	<p>This command sets the particular lane number for the injection purpose.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTL</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:LANE &lt;wsp&gt; &lt;Lane&gt;, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which injection is to be done.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the OTL per lane status.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTL:LANE 1, ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALANes</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:LANE?**

<b>Description</b>	<p>This query returns the particular lane number for the injection purpose.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTL</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:LANE? &lt;wsp&gt; &lt;Lane&gt;</code>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which injection is to be done.</p>
<b>Response Syntax</b>	<code>&lt;Set&gt;</code>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of OTL lane number.</p> <p>1, particular lane is enabled.</p> <p>0, particular lane is disabled.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:OTL:LANE? 1</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALANes?</code>

---

## SCPI Command Reference

### *Alarms/Errors*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALANes**

<b>Description</b>	<p>This command sets all lanes for the injection purpose.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTL</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALANes &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the all lane status.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:OTL:ALAN ON</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:OTN:OTL:LANE</code>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALANes?**

---

<b>Description</b>	<p>This query returns status of all lanes for the injection purpose.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; OTL</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:ALANes?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of OTL all lanes.</p> <p>1, all lanes are enabled.</p> <p>0, all lanes are disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:ALAN?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTL:LANE?

---

## SCPI Command Reference

### Alarms/Errors

---

<b>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:TYPE</b>	
<b>Description</b>	<p>This command selects the manual type of global EOTN error.</p> <p>At *RST condition, the value is set to Inv. Flag.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:TYPE &lt;wsp&gt;INVALFLAG   MSEQV</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: INVALFLAG   MSEQV</p> <p>Selects the type of global EOTN manual error.</p> <p>INVALFLAG:s INVALFLAG</p> <p>MSEQV: MSEQV</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:MAN:TYPE MSEQV</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:TYPE</pre>

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:TYPE?**

---

<b>Description</b>	<p>This query returns the manual type of global EOTN error.</p> <p>At *RST condition, the value is set to Inv. Flag.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of manual global EOTN error.</p> <p>INVALFLAG, the EOTN manual error type is selected as INVALFLAG.</p> <p>MSEQV, the EOTN manual error type is selected as MSEQV.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:MAN:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:TYPE?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:AMOut**

<b>Description</b>	<p>This command sets the amount for the global EOTN error.</p> <p>At *RST condition, the value is set to 1.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:AMOut &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the global EOTN error amount value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:MAN:AMO 50</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:AMOut</p>

---

<b>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:AMOUnt?</b>	
<b>Description</b>	<p>This query returns the amount value for the global EOTN error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:AMOUnt?</code> <wsp>MAXimum   MINimum
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of global EOTN error.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:MAN:AMO?</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:AMOUnt?</code>

---

## SCPI Command Reference

### *Alarms/Errors*

---

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:INJect**

---

<b>Description</b>	<p>This command injects the global EOTN error.</p> <p>This command is not associated with any *RST condition.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:MAN:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:INJect

---



---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:TYPE**

---

<b>Description</b>	<p>This command selects the automated type of global EOTN error.</p> <p>At *RST condition, the value is set to Inv. Flag.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:TYPE &lt;wsp&gt;INVALFLAG   MSEQV</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: INVALFLAG   MSEQV</p> <p>Selects the type of EOTN automated error.</p> <p>INVALFLAG: INVALFLAG</p> <p>MSEQV: MSEQV</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:AUT:TYPE MSEQV</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:TYPE</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

<b>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:TYPE?</b>	
<b>Description</b>	This query returns the automated type of global EOTN error. At *RST condition, the value is set to Inv. Flag. Navigation Path: Test > Results > Alarms/Errors > Global Injection > Transcoding
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<b>Error:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the selected type of automated global EOTN error.
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:AUT:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:TYPE?

---

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:RATE**

---

<b>Description</b>	<p>This command sets the automated rate for the global EOTN error.</p> <p>At *RST condition, the value is set to 1.0E-04.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the global EOTN error rate value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:AUT:RATE 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:RATE</p>

---

## SCPI Command Reference

### Alarms/Errors

---

<b>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:RATE?</b>	
<b>Description</b>	<p>This query returns the automated rate value for the global EOTN error.</p> <p>At *RST condition, the value is set to 1.0E-04.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:RATE? &lt;wsp&gt;MAXimum   MINimum</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of automated global EOTN error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:AUT:RATE?</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:RATE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:CONTInuous**

---

<b>Description</b>	<p>This command sets the continuous mode status of the global EOTN automated error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the global EOTN continuous mode status value.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:AUT:CONT ON</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:CONTInuous</code>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### **:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:CONTInuous?**

---

<b>Description</b>	This query returns the continuous mode status of the global EOTN automated error. At *RST condition, this value is set to OFF. Navigation Path: Test > Results > Alarms/Errors > Global Injection > Transcoding
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<b>Set:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated global EOTN error continuous mode. 1, continuous mode is enabled. 0, continuous mode is disabled.
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:AUT:CONT?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:CONTInuous?

---

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated**

---

<b>Description</b>	<p>This command sets the global EOTN automated error generation status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the global EOTN error generation status value.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:AUT ON</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

<b>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated?</b>	
<b>Description</b>	<p>This query returns the global EOTN automated error generation status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of global EOTN automated error generation.</p> <p>1, error generation is enabled.</p> <p>0, error generation is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:GLOB:AUT?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated?

---



**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:MANual:TYPE**

<b>Description</b>	<p>This command selects the manual type of EOTN error.</p> <p>At *RST condition, the value is set to Invalid Flag.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:MANual:TYPE &lt;wsp&gt;PCSBIP8MASK   OTNBIP8</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCSBIP8MASK   OTNBIP8</p> <p>Selects the type of EOTN manual error.</p> <p>OTNBIP8: OTN BIP-8</p> <p>PCSBIP8MASK: PCS BIP-8 Mask</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:EOTN:TRAN:ERR:MAN:TYPE PCSBIP8MASK</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:EOTN:TRANScode:ERRor:GLOBal:MANual:TYPE</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type of EOTN error.</p> <p>At *RST condition, the value is set to Invalid Flag.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of manual EOTN error.</p> <p>OTN BIP-8, OTN BIP-8 is selected as the EOTN manual error type.</p> <p>PCS BIP-8 Mask, PCS BIP-8 Mask is selected as the EOTN manual error type.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:MAN:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:TYPE?

---

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:AMOut**

---

<b>Description</b>	<p>This command sets the amount for the EOTN error.</p> <p>At *RST condition, the value is set to 1.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:AMOut &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the EOTN error amount value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:EOTN:TRAN:ERR:MAN:AMO 25</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:AMOut</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:AMOut?**

<b>Description</b>	<p>This query returns the amount value for the EOTN error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:AMOut? &lt;wsp&gt;MAXimum   MINimum</p>
<b>Response Syntax</b>	<p>&lt;Amount&gt;</p>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of EOTN error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:EOTN:TRAN:ERR:MAN:AMO?</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:AMOut?</p>

<b>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:INJect</b>	
<b>Description</b>	<p>This command injects the EOTN error.</p> <p>This is an event and not associated with any *RST condition.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:MANual:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:MAN:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:MANual:INJect

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the automated type of EOTN error.</p> <p>At *RST condition, the value is set to Invalid Flag.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:TYPE &lt;wsp&gt;PCSBIP8MASK   OTNBIP8</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCSBIP8MASK   OTNBIP8</p> <p>Selects the type of EOTN automated error.</p> <p>OTNBIP8, OTN BIP-8 is the EOTN automated error type selected.</p> <p>PCSBIP8MASK, PCS BIP-8 Mask is the EOTN automated error type selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:EOTN:TRAN:ERR:AUT:TYPE OTNBIP8</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:TYPE</p>

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:TYPE?**

---

<b>Description</b>	<p>This query returns the automated type of EOTN error.</p> <p>At *RST condition, the value is set to Invalid Flag.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of automated EOTN error.</p> <p>OTNBIP8, the EOTN automated error type is selected as OTN BIP-8.</p> <p>PCSBIP8MASK, the EOTN automated error type is selected as PCS BIP-8 Mask.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:EOTN:TRAN:ERR:AUT:TYPE OTNBIP8</p> <p>SOUR:DATA:TEL:EOTN:TRAN:ERR:AUT:TYPE?</p> <p>Returns: OTNBIP8</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:TYPE?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:RATE**

<b>Description</b>	<p>This command sets the automated rate for the EOTN error.</p> <p>At *RST condition, the value is set to 1.0E-04.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the EOTN error rate value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:EOTN:TRAN:ERR:AUT:RATE 1.0E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:RATE</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:RATE?**

---

<b>Description</b>	<p>This query returns the automated rate value for the EOTN error.</p> <p>At *RST condition, the value is set to 1.0E-04.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:RATE? &lt;wsp&gt;MAXimum   MINimum</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of automated EOTN error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:EOTN:TRAN:ERR:AUT:RATE?</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:RATE?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:CONTInuous**

<b>Description</b>	<p>This command sets the continuous mode status of the EOTN automated error.</p> <p>At *RST condition, the value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the EOTN continuous mode status value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:EOTN:TRAN:ERR:AUT:CONT ON</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:CONTInuous</p>

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:CONTInuous?**

<b>Description</b>	<p>This query returns the continuous mode status of the EOTN automated error.</p> <p>At *RST condition, the value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated EOTN error continuous mode.</p> <p>1, continuous mode is enabled.</p> <p>0, continuous mode is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:AUT:CONT?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated:CONTInuous?

---

## SCPI Command Reference

### *Alarms/Errors*

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated**

<b>Description</b>	<p>This command sets the EOTN automated error generation status.</p> <p>At *RST condition, the value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the EOTN error generation status value.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:EOTN:TRAN:ERR:AUT ON</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated</code>

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated?**

<b>Description</b>	<p>This query returns the EOTN automated error generation status.</p> <p>At *RST condition, the value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of EOTN automated error generation.</p> <p>1, error generation is enabled.</p> <p>0, error generation is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:AUT?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:GLOBal:AUTomated?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:LANE

---

<b>Description</b>	<p>This command sets the particular lane number for the injection purpose.</p> <p>At *RST condition, the value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:LANE <wsp> <Lane>, ON   OFF
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which injection is to be done for EOTN lane error.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the OTL per lane status for EOTN error.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:LANE 1, ON
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:ALANes

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:LANE?**

<b>Description</b>	<p>This query returns the particular lane number for the injection purpose.</p> <p>At *RST condition, the value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:LANE? <wsp> <Lane>
<b>Parameter(s)</b>	<p><b>Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the lane number for which injection is to be done for EOTN lane error.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of EOTN particular lane.</p> <p>1, particular lane is enabled.</p> <p>0, particular lane is disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:LANE? 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:ALANes?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:ALANes**

<b>Description</b>	<p>This command sets all lanes for the injection purpose.</p> <p>At *RST condition, the value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:ALANes &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the all lane status for EOTN lane error.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:EOTN:TRAN:ERR:ALAN ON</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:LANE</code>

---



**:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:ALANes?**

---

<b>Description</b>	<p>This query returns all lanes for the injection purpose.</p> <p>At *RST condition, the value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Transcoding</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:ALANes?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of EOTN all lanes.</p> <p>1, all lanes are enabled.</p> <p>0, all lanes are disabled.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:EOTN:TRAN:ERR:ALAN?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:EOTN:TRANscode:ERRor:LANE?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### **:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ALARm:HISTory?**

---

<b>Description</b>	<p>This query returns the history status for EOTN alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Transcoding &gt; Alarms</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ALARm:HISTory? &lt;wsp&gt;LOBL1027B   HIBER1027B   LOAML1027B</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOBL1027B   HIBER1027B   LOAML1027B</p> <p>Selects the alarm for the history status.</p> <p>LOBL1027B: LOBL1027B for the history status of alarm.</p> <p>LOAM1027B: LOAM1027B for the history status of alarm.</p> <p>HIBER1027B: HIBER1027B for the history status of alarm.</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of EOTN alarm.</p> <p>PRESENT, indicates that at least one physical error is present.</p> <p>ABSENT, indicates that there is no physical error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:EOTN:XCOD:TRAN:RX:ALAR:HIST? LOBL1027B</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:HISTory?</p>

---

**:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ALARm:CURRent?**

<b>Description</b>	<p>This query returns Current Status for EOTN alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Transcoding &gt; Alarms</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ALARm:CURRent? &lt;wsp&gt;LOBL1027B   HIBER1027B   LOAML1027B</pre>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOBL1027B   HIBER1027B   LOAML1027B</p> <p>Selects the alarm for the CURRENT status.</p> <p>LOBL1027B: LOBL1027B for the current status of alarm.</p> <p>LOAM1027B: LOAM1027B for the current status of alarm.</p> <p>HIBER1027B: HIBER1027B for the current status of alarm.</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of EOTN alarm.</p> <p>PRESENT, indicates that at least one physical error is present.</p> <p>ABSENT, indicates that there is no physical error.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:XCOD:TRAN:RX:ALAR:CURR? LOBL1027B
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ALARm:SEConds?**

<b>Description</b>	<p>This query returns seconds for EOTN alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Transcoding &gt; Alarms</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ALARm:SEConds? &lt;wsp&gt;LOBL1027B   HIBER1027B   LOAML1027B</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOBL1027B   HIBER1027B   LOAML1027B</p> <p>Selects the alarm for the seconds.</p> <p>LOBL1027B: LOBL1027B for the history status of alarm.</p> <p>LOAM1027B: LOAM1027B for the history status of alarm.</p> <p>HIBER1027B: HIBER1027B for the history status of alarm.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the seconds value of EOTN alarm.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:EOTN:XCOD:TRAN:RX:ALAR:SEC? LOBL1027B</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:SEConds?</p>

**:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:HISTory?**

<b>Description</b>	<p>This query returns the history status of the EOTN error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors (Ethernet) &gt; Ethernet &gt; Transcoding &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:HISTory? <wsp>POSV   SEQV   MSEQV   INVALFLAG   PCSBIP8MASK   OTNBIP8
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: POSV   SEQV   MSEQV   INVALFLAG   PCSBIP8MASK   OTNBIP8</p> <p>Sets the history status of the EOTN error.</p> <p>POSV: POSV for history status of EOTN error.</p> <p>SEQV: SEQV for history status of EOTN error.</p> <p>INAVLFLAG: INVALFLAG for history status of EOTN error.</p> <p>MSEQV: MSEQV for history status of EOTN error.</p> <p>PCSBIP8MASK: PCSBIP8MASK for history status of EOTN error.</p> <p>OTNBIP8: OTNBIP8 for history status of EOTN error.</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the History status of EOTN error.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:XCOD:TRAN:RX:ERR:HIST? PCSBIP8MASK
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ALARm:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:CURRent?**

<b>Description</b>	<p>This query returns the current status of the EOTN error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet (Ethernet) &gt; Transcoding &gt; Errors</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:CURRent? &lt;wsp&gt;POSV   SEQV   MSEQV   INVALFLAG   PCSBIP8MASK   OTNBIP8</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: POSV   SEQV   MSEQV   INVALFLAG   PCSBIP8MASK   OTNBIP8</p> <p>Sets the current status of the EOTN error.</p> <p>POSV: POSV for current status of EOTN error.</p> <p>SEQV: SEQV for current status of EOTN error.</p> <p>INAVLFLAG: INVALFLAG for current status of EOTN error.</p> <p>MSEQV: MSEQV for current status of EOTN error.</p> <p>PCSBIP8MASK: PCSBIP8MASK for current status of EOTN error.</p> <p>OTNBIP8: OTNBIP8 for current status of EOTN error.</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of EOTN error.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:EOTN:XCOD:TRAN:RX:ERR:CURR? PCSBIP8MASK</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ALARm:CURRent?</p>

**:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:SEConds?**

<b>Description</b>	<p>This query returns the seconds value of the EOTN error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Transcoding &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:SEConds? <wsp>POSV   SEQV   MSEQV   INVALFLAG   PCSBIP8MASK   OTNBIP8
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: POSV   SEQV   MSEQV   INVALFLAG   PCSBIP8MASK   OTNBIP8</p> <p>Sets the seconds of the EOTN error.</p> <p>POSV: POSV for seconds of EOTN error.</p> <p>SEQV: SEQV for seconds of EOTN error.</p> <p>INAVLFLAG: INVALFLAG for seconds of EOTN error.</p> <p>MSEQV: MSEQV for seconds of EOTN error.</p> <p>PCSBIP8MASK: PCSBIP8MASK for seconds of EOTN error.</p> <p>OTNBIP8: OTNBIP8 for seconds of EOTN error.</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the seconds value of EOTN Error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:XCOD:TRAN:RX:ERR:SEC? PCSBIP8MASK
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ALARm:SEConds?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:COUNT?**

<b>Description</b>	<p>This query returns the count of the EOTN error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Transcoding &gt; Errors</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:COUNT? &lt;wsp&gt;POSV   SEQV   MSEQV   INVALFLAG   PCSBIP8MASK   OTNBIP8</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: POSV   SEQV   MSEQV   INVALFLAG   PCSBIP8MASK   OTNBIP8</p> <p>Sets the count of the EOTN error.</p> <p>POSV: POSV for count of EOTN error.</p> <p>SEQV: SEQV for count of EOTN error.</p> <p>INAVLFLAG: INVALFLAG for count of EOTN error.</p> <p>MSEQV: MSEQV for count of EOTN error.</p> <p>PCSBIP8MASK: PCSBIP8MASK for count of EOTN error.</p> <p>OTNBIP8: OTNBIP8 for count of EOTN error.</p>
<b>Response Syntax</b>	<p>&lt;Count&gt;</p>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of EOTN Error.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:EOTN:XCOD:TRAN:RX:ERR:COUN? PCSBIP8MASK</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:HISTory?</p>



**:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:RATE?**

<b>Description</b>	<p>This query returns the rate value of the EOTN error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Transcoding &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:RATE? <wsp>POSV   SEQV   MSEQV   INVALFLAG   PCSBIP8MASK   OTNBIP8
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: POSV   SEQV   MSEQV   INVALFLAG   PCSBIP8MASK   OTNBIP8</p> <p>Sets the rate of the EOTN error.</p> <p>POSV: POSV for rate of EOTN error.</p> <p>SEQV: SEQV for rate of EOTN error.</p> <p>INAVLFLAG: INVALFLAG for rate of EOTN error.</p> <p>MSEQV: MSEQV for rate of EOTN error.</p> <p>PCSBIP8MASK: PCSBIP8MASK for rate of EOTN error.</p> <p>OTNBIP8: OTNBIP8 for rate of EOTN error.</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate value of EOTN Error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:XCOD:TRAN:RX:ERR:RATE? PCSBIP8MASK
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:XCODing:TRANscode:RX:ERRor:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:HISTory?

<b>Description</b>	<p>This query returns the history state within which the EOTN physical error occurred for Each lane.</p> <p>For EOTN in the ""Modify Structure"" block, in the ""Client"" dropdown, set ""Ethernet"" as the value, and for interface set ""OTU3"".</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: EOTN Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; PCS Lanes &gt; Errors (PCSBIP8MASK, OTNBIP8)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:HISTory? <wsp> <lane>, PCSBIP8MASK   OTNBIP8
<b>Parameter(s)</b>	<p><b>lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCSBIP8MASK   OTNBIP8</p> <p>Selects the type of EOTN error for each lane.</p> <p>PCSBIP8MASK: PCSBIP8MASK</p> <p>OTNBIP8: OTNBIP8</p>
<b>Response Syntax</b>	<History>

---

**:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Reports the history status of EOTN error for per lane configuration. PRESENT, indicates that at least one alarm has occurred. ABSENT, indicates that no alarm occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:ERR:PHYS:HIST? 1, OTNBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE?

---

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELEcom:EOTN:ERROr:PHYSical:CURRent?

<b>Description</b>	<p>This query returns the current state within which EOTN physical error occurred for Each lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: EOTN Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; PCS Lanes &gt; Errors (PCSBIP8MASK, OTNBIP8)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:EOTN:ERROr:PHYSical:CURRent? <wsp> <lane>, PCSBIP8MASK   OTNBIP8
<b>Parameter(s)</b>	<p><b>lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCSBIP8MASK   OTNBIP8</p> <p>Selects the type of EOTN error for each lane.</p> <p>PCSBIP8MASK: PCSBIP8MASK</p> <p>OTNBIP8: OTNBIP8</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the current status of EOTN error for per lane configuration.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that there is no alarm.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:ERR:PHYS:CURR? 1, OTNBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:ERROr:PHYSical:RATE?

**:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the EOTN physical error occurred for Each lane.</p> <p>For EOTN in the ""Modify Structure"" block, in the ""Client"" dropdown, set ""Ethernet"" as the value, and for interface set ""OTU3"".</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: EOTN Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; PCS Lanes &gt; Errors (PCSBIP8MASK, OTNBIP8)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:SEConds? <wsp> <lane>, PCSBIP8MASK   OTNBIP8
<b>Parameter(s)</b>	<p><b>lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCSBIP8MASK   OTNBIP8</p> <p>Selects the type of EOTN error for each lane.</p> <p>PCSBIP8MASK: PCSBIP8MASK</p> <p>OTNBIP8: OTNBIP8</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the number of seconds of EOTN error for per lane configuration.</p>
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:ERR:PHYS:SEC? 1, OTNBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:COUNT?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:EOTN:ERRor:PHYSical:COUnT?

---

<b>Description</b>	<p>This query returns the count within which the EOTN physical error occurred for Each lane.</p> <p>For EOTN in the ""Modify Structure"" block, in the ""Client"" dropdown, set ""Ethernet"" as the value, and for interface set ""OTU3"".</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: EOTN Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; PCS Lanes &gt; Errors (PCSBIP8MASK, OTNBIP8)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:EOTN:ERRor:PHYSical:COUnT? <wsp> <lane>, PCSBIP8MASK   OTNBIP8
<b>Parameter(s)</b>	<p><b>lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCSBIP8MASK   OTNBIP8</p> <p>Selects the type of EOTN error for each lane.</p> <p>PCSBIP8MASK: PCSBIP8MASK</p> <p>OTNBIP8: OTNBIP8</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the count of EOTN error for per lane configuration.</p>
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:ERR:PHYS:COUN? 1, OTNBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:EOTN:ERRor:PHYSical:CURREnt?

---

---

**:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE?**

---

<b>Description</b>	<p>This query returns the rate at which EOTN physical error occurred for Each lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: EOTN Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; PCS Lanes &gt; Errors (PCSBIP8MASK, OTNBIP8)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE? &lt;wsp&gt; &lt;lane&gt;, PCSBIP8MASK   OTNBIP8</p>
<b>Parameter(s)</b>	<p><b>lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane number.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCSBIP8MASK   OTNBIP8</p> <p>Selects the type of EOTN error for each lane.</p> <p>PCSBIP8MASK: PCSBIP8MASK</p> <p>OTNBIP8: OTNBIP8</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the rate of EOTN error for per lane configuration.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:EOTN:ERR:PHYS:RATE? 1, OTNBIP8</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:HISTory?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:COUnt:TOTal?

---

<b>Description</b>	<p>This query returns the Total count within which the EOTN physical error occurred for Each lane.</p> <p>For EOTN in the ""Modify Structure"" block, in the ""Client"" dropdown, set ""Ethernet"" as the value, and for interface set ""OTU3"".</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: EOTN Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; PCS Lanes &gt; Errors (PCSBIP8MASK, OTNBIP8) TOTAL</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:COUnt:TOTal? <wsp>PCSBIP8MASK   OTNBIP8
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCSBIP8MASK   OTNBIP8</p> <p>Selects the type of EOTN error for each lane.</p> <p>PCSBIP8MASK: PCSBIP8MASK</p> <p>OTNBIP8: OTNBIP8</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the count of EOTN error for Total lane configuration.</p>
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:ERR:PHYS:COUN:TOTal? OTNBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:CURRent?

---



**:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE:TOTal?**

<b>Description</b>	<p>This query returns the total rate at which EOTN physical error occurred for Each lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: EOTN Test &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; PCS Lanes &gt; Errors (PCSBIP8MASK, OTNBIP8) TOTAL</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE:TOTal? &lt;wsp&gt;PCSBIP8MASK   OTNBIP8</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCSBIP8MASK   OTNBIP8</p> <p>Selects the type of EOTN error for each lane.</p> <p>PCSBIP8MASK: PCSBIP8MASK</p> <p>OTNBIP8: OTNBIP8</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Reports the rate of EOTN error for total lane configuration.</p>
<b>Example(s)</b>	FETC:DATA:TEL:EOTN:ERR:PHYS:RATE:TOTal? OTNBIP8
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:EOTN:ALARm:LINK?**

<b>Description</b>	<p>This query returns the status of Ethernet alarms.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; 1GB/10GB/Ethernetflex &gt; Interface</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; 10/100/1000M Electrical/10GE LAN/10GE WAN/100M Optical/1GE Optical &gt; Interface</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:EOTN:ALARm:LINK? &lt;wsp&gt;LDOWN   LFAR   LFAD   RFAult</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LDOWN   LFAR   LFAD   RFAult</p> <p>Selects the alarm type whose status is to be retrieved.</p> <p>LDOWN: Link Down</p> <p>LFAR: Fault Received</p> <p>LFAD: Local Fault Detected</p> <p>RFAult: RFAult</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Reports the status of Ethernet alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETHernet:EOTN:ALAR:LINK? RFAULT</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:PATTer:n:ALARm:SYNCh?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE?**

<b>Description</b>	<p>This query returns the type of Optical Payload Unit (OPU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU 1/2 E) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Optical Payload Unit (OPU) alarm.</p> <p>OMSIM, OPU-OMSIM is selected as the alarm type.</p> <p>OAIS, OPU-OAIS is selected as the alarm type.</p> <p>OCSF, OPU-OCSF is selected as the alarm type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OAIS</p> <p>SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE?</p> <p>Returns: OAIS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E**

<b>Description</b>	<p>This command enables or disables the status of Optical Payload Unit (OPU) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU 1/2 E) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the OPU (Optical Payload Unit) alarm generation.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OAIS SOUR:DATA:TEL:OTN:ALAR:OPU1:E ON SOUR:DATA:TEL:OTN:ALAR:OPU1:E? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE?</pre>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E?**

<b>Description</b>	<p>This query returns the status of Optical Payload Unit (OPU) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (OPU 1/2 E) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Optical Payload Unit (OPU) alarm generation.</p> <p>1, alarm generation is enabled.</p> <p>0, alarm generation is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OAIS</p> <p>SOUR:DATA:TEL:OTN:ALAR:OPU1:E ON</p> <p>SOUR:DATA:TEL:OTN:ALAR:OPU1:E?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELeom:OTN:ALARm:OPU[1..n]:E:HISTory?

---

<b>Description</b>	<p>This query returns the history status of the Optical Payload Unit (OPU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU 1/2 E&gt; OPU 1/2E</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELeom:OTN:ALARm:OPU[1..n]:E:HISTory? &lt;wsp&gt;OAIS   OPLM   OMSim   OCSF</code>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OPLM   OMSim   OCSF</p> <p>Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OAIS: OPU-AIS (Optical Payload Unit Alarm Indication Signal)</p> <p>OPLM: OPU-PLM (Optical Payload Unit Payload Mismatch)</p> <p>OMSim: OPU-MSIM (Optical Payload Unit Multiplex Structure Identifier Mismatch)</p> <p>OCSF: OPU-CSF (Optical Payload Unit Client Signal Fail)</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of the Optical Payload Unit (OPU) alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:OTN:ALAR:OPU1:E:HIST? OPLM</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELeom:OTN:ALARm:OPU[1..n]:CURRent?</code>

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OPU[1..n]:E:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the Optical Payload Unit (OPU) alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU 1/2 E &gt; OPU 1/2E</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OPU[1..n]:E:SEConds? <wsp>OAIS   OPLM   OMSim   OCSF
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OPLM   OMSim   OCSF</p> <p>Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OAIS: OPU-AIS (Optical Payload Unit Alarm Indication Signal)</p> <p>OPLM: OPU-PLM (Optical Payload Unit Payload Mismatch)</p> <p>OMSim: OPU-MSIM (Optical Payload Unit Multiplex Structure Identifier Mismatch)</p> <p>OCSF: OPU-CSF (Optical Payload Unit Client Signal Fail)</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of the Optical Payload Unit (OPU) alarm.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ALAR:OPU1:E:SEC? OPLM
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ALARm:OPU[1..n]:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OPU[1..n]:E:CURRent?**

<b>Description</b>	<p>This query returns the current status of the Optical Payload Unit (OPU) alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; Alarms/Errors &gt; ODU 1/2 E&gt; OPU 1/2E</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:OTN:ALARm:OPU[1..n]:E:CURRent? &lt;wsp&gt;OAIS   OPLM   OMSim   OCSF</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OAIS   OPLM   OMSim   OCSF</p> <p>Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OAIS: OPU-AIS (Optical Payload Unit Alarm Indication Signal)</p> <p>OPLM: OPU-PLM (Optical Payload Unit Payload Mismatch)</p> <p>OMSim: OPU-MSIM (Optical Payload Unit Multiplex Structure Identifier Mismatch)</p> <p>OCSF: OPU-CSF (Optical Payload Unit Client Signal Fail)</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of the Optical Payload Unit (OPU) alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that there is no alarm.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ALAR:OPU1:E:CURR? OPLM</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:ALARm:OPU[1..n]:HISTory?</p>



**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead**

<b>Description</b>	<p>This command sets the Optical Payload Unit (OPU) E overhead byte values for the transmitter.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead <wsp>JC6   JC5   JC4   JC3   JC2   JC1   RES416   JC116   NJO416   OMFI, <Value>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: JC6   JC5   JC4   JC3   JC2   JC1   RES416   JC116   NJO416   OMFI</p> <p>Selects OPU overhead bytes.</p> <p>JC6</p> <p>JC5</p> <p>JC4</p> <p>JC3</p> <p>JC2</p> <p>JC1</p> <p>RES416</p> <p>JC116</p> <p>NJO416</p> <p>OMFI</p> <p><b>Value:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Optical Payload Unit (OPU) overhead byte values.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OH:OPU1:E:OVER JC6, #HF6
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:HISTory?

---

<b>Description</b>	<p>This query returns the history status of the Client Frequency Alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Results &gt; Alarms/Errors &gt; Bert &gt; Client Frequency</p> <p>Navigation Path for 1GbE client: Test(OTN Bert) &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Ethernet &gt; Client frequency</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:HISTory? <wsp>CFRequency
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CFRequency</p> <p>Selects the type of alarm for history status</p> <p>CFRequency: CFRequency</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of the Alarm</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:COFF:ALAR:HIST? CFR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm: SEConds?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:CURRent?**

<b>Description</b>	<p>This query returns the current status of the Client Frequency Alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Results &gt; Alarms/Errors &gt; Bert &gt; Client Frequency</p> <p>Navigation Path for 1GbE client: Test(OTN Bert) &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Ethernet &gt; Client frequency</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:CURRent? <wsp>CFRequency
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CFRequency</p> <p>Selects the type of alarm for current status</p> <p>CFRequency: CFRequency</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of the Alarm</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:COFF:ALAR:CURR? CFR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm: SEConds?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds for the Client Frequency alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Results &gt; Alarms/Errors &gt; Bert &gt; Client Frequency</p> <p>Navigation Path for 1GbE client: Test(OTN Bert) &gt; Results &gt; Alarms/Errors &gt; Ethernet &gt; Ethernet &gt; Client frequency</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:SEConds? <wsp>CFRequency
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CFRequency</p> <p>Selects the type of alarm for seconds value</p> <p>CFRequency: CFRequency</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds for the alarm.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:COFF:ALAR:SEC? CFR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm: HISTory?

---

**:FETCh[1..n]:DATA:TELeCom:ALARm:HISTory?**

<b>Description</b>	<p>This query returns the history status of all alarms related to the test such as Port, Ethernet, Pattern and Other. It also returns the combined status for all the tests.</p> <p>At *RST condition, this value is device dependent.</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELeCom:ALARm:HISTory? &lt;wsp&gt;PATTERN   PORT   CLOCK   PCS   OTL   OPU0   OPU1   OPU2   OPU3   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   OPUFLEX   ODUFLEX   ODUFLEXTCM   EOTNPCS   GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM[, &lt;Channel&gt;]</pre>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PATTERN   PORT   CLOCK   PCS   OTL   OPU0   OPU1   OPU2   OPU3   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   OPUFLEX   ODUFLEX   ODUFLEXTCM   EOTNPCS   GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM</p> <p>Selects all alarms related to the selected test and all the tests.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<History>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:ALARm:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the alarmed history status of the selected test and all the tests.
<b>Example(s)</b>	FETC:DATA:TEL:ALAR:HIST? PORT FETC:DATA:TEL:ALAR:HIST? ODU0, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ALARm:CURRent? FETCh[1..n]:DATA:TELecom:ALARm:SECOnds?

---

**:FETCh[1..n]:DATA:TELEcom:ALARm:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which all alarms related to the test such as Port, Ethernet, Pattern and Other. It also returns the combined status for all the tests.</p> <p>At *RST condition, this value is device dependent.</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ALARm:SEConds? &lt;wsp&gt;PATTERN   PORT   CLOCK   PCS   OTL   OPU0   OPU1   OPU2   OPU3   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   OPUFLEX   ODUFLEX   ODUFLEXTCM   EOTNPCS   GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PATTERN   PORT   CLOCK   PCS   OTL   OPU0   OPU1   OPU2   OPU3   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   OPUFLEX   ODUFLEX   ODUFLEXTCM   EOTNPCS   GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM</p> <p>Selects all alarms related to the selected test and all the tests.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:ALARm:SEConds?**

<b>Response(s)</b>	<b>Seconds:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the number of seconds within which the selected test alarms and all the test alarms occurred.
<b>Example(s)</b>	FETC:DATA:TEL:ALAR:SEC? PORT FETC:DATA:TEL:ALAR:SEC? ODU0, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ALARm:HISTory? FETCh[1..n]:DATA:TELEcom:ALARm:CURREnt?

---



**:FETCh[1..n]:DATA:TELEcom:ALARm:CURRent?**

<b>Description</b>	<p>This query returns the current status of all alarms related to the test such as Port, Ethernet, Pattern and Other. It also returns the combined status for all the tests.</p> <p>At *RST condition, this value is device dependent.</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:ALARm:CURRent? &lt;wsp&gt;PATTERN   PORT   CLOCK   PCS   OTL   OPU0   OPU1   OPU2   OPU3   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   OPUFLEX   ODUFLEX   ODUFLEXTCM   EOTNPCS   GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM[, &lt;Channel&gt;]</pre>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PATTERN   PORT   CLOCK   PCS   OTL   OPU0   OPU1   OPU2   OPU3   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   OPUFLEX   ODUFLEX   ODUFLEXTCM   EOTNPCS   GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM</p> <p>Selects all alarms related to the selected test and all the tests.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Current>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:ALARm:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the current alarmed status of the selected test and all the tests.
<b>Example(s)</b>	FETC:DATA:TEL:ALAR:CURR? PORT FETC:DATA:TEL:ALAR:CURR? ODU0, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ALARm:HISTory? FETCh[1..n]:DATA:TELEcom:ALARm:SECOnds?

---

**:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?**

<b>Description</b>	<p>This query returns the history status of all errors related to the test such as Ethernet, Pattern and Other. It also returns the combined status for all the tests.</p> <p>At *RST condition, this value is device dependent.</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory? &lt;wsp&gt;PATTERN   PCS   ETHERNET   IPUDP   OTL   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   FEC   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   EOTNPCS   QOSMETRICS   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   ODUFLEX   ODUFLEXTCM     GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM[, &lt;Channel&gt;]</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PATTERN   PCS   ETHERNET   IPUDP   OTL   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   FEC   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   EOTNPCS   QOSMETRICS   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   ODUFLEX   ODUFLEXTCM     GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM</p> <p>Selects all errors related to the selected test and all the tests.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<pre>&lt;History&gt;</pre>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the errored history status of the selected test and all the tests.
<b>Example(s)</b>	FETC:DATA:TEL:ERR:HIST? ODU3 FETC:DATA:TEL:ERR:HIST? ODU3, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ERRor:CURRent? FETCh[1..n]:DATA:TELEcom:ERRor:SEConds? FETCh[1..n]:DATA:TELEcom:ERRor:RATE? FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?

---

**:FETCh[1..n]:DATA:TELeCom:ERRor:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which all errors related to the test such as Ethernet, Pattern and Other occurred. It also returns the combined status for all the tests.</p> <p>At *RST condition, this value is device dependent.</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELeCom:ERRor:SEConds? &lt;wsp&gt;PATTERN   PCS   ETHERNET   IPUDP   OTL   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   FEC   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   EOTNPCS   QOSMETRICS   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   ODUFLEX   ODUFLEXTCM     GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM[, &lt;Channel&gt;]</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PATTERN   PCS   ETHERNET   IPUDP   OTL   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   FEC   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   EOTNPCS   QOSMETRICS   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   ODUFLEX   ODUFLEXTCM     GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM</p> <p>Selects all errors related to the selected test and all the tests.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Seconds>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:ERRor:SEConds?**

<b>Response(s)</b>	<b>Seconds:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the number of seconds within which the selected test errors and all the test errors occurred.
<b>Example(s)</b>	FETC:DATA:TEL:ERR:SEC? ODU3 FETC:DATA:TEL:ERR:SEC? ODU3, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ERRor:HISTory? FETCh[1..n]:DATA:TELEcom:ERRor:CURREnt? FETCh[1..n]:DATA:TELEcom:ERRor:RATE? FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?

---

**:FETCh[1..n]:DATA:TELecom:ERRor:CURRent?**

<b>Description</b>	<p>This query returns the current status of all errors related to the test such as Ethernet, Pattern and Other. It also returns the combined status for all the tests.</p> <p>At *RST condition, this value is device dependent.</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ERRor:CURRent? &lt;wsp&gt;PATTERN   PCS   ETHERNET   IPUDP   OTL   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   FEC   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   EOTNPCS   QOSMETRICS   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   ODUFLEX   ODUFLEXTCM     GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PATTERN   PCS   ETHERNET   IPUDP   OTL   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   FEC   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   EOTNPCS   QOSMETRICS   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   ODUFLEX   ODUFLEXTCM     GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM</p> <p>Selects all errors related to the selected test and all the tests.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:ERRor:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the current errored status of the selected test and all the tests.
<b>Example(s)</b>	FETC:DATA:TEL:ERR:CURR? ODU3 FETC:DATA:TEL:ERR:CURR? ODU3, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ERRor:HISTory? FETCh[1..n]:DATA:TELEcom:ERRor:SECOnds? FETCh[1..n]:DATA:TELEcom:ERRor:RATE? FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?

---



**:FETCh[1..n]:DATA:TELEcom:ERRor:COUNt?**

<b>Description</b>	<p>This query returns the count of all errors related to the test such as Ethernet, Pattern and Other. It also returns the combined status for all the tests.</p> <p>At *RST condition, this value is device dependent.</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:ERRor:COUNt? &lt;wsp&gt;PATTERN   PCS   ETHERNET   IPUDP   OTL   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   FEC   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   EOTNPCS   QOSMETRICS   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   ODUFLEX   ODUFLEXTCM     GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM[, &lt;Channel&gt;]</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PATTERN   PCS   ETHERNET   IPUDP   OTL   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   FEC   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   EOTNPCS   QOSMETRICS   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   ODUFLEX   ODUFLEXTCM     GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM</p> <p>Selects all errors related to the selected test and all the tests.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Count>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?

<b>Response(s)</b>	<b>Count:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the error count for the selected test errors and all the test errors.
<b>Example(s)</b>	FETC:DATA:TEL:ERR:COUN? ODU3 FETC:DATA:TEL:ERR:COUN? ODU3, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ERRor:HISTory? FETCh[1..n]:DATA:TELEcom:ERRor:CURREnt? FETCh[1..n]:DATA:TELEcom:ERRor:SECConds? FETCh[1..n]:DATA:TELEcom:ERRor:RATE?

---

**:FETCh[1..n]:DATA:TELEcom:ERRor:RATE?**

<b>Description</b>	<p>This query returns the current rate of all errors related to the test such as Ethernet, Pattern and Other. It also returns the combined status for all the tests.</p> <p>At *RST condition, this value is device dependent.</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:ERRor:RATE? &lt;wsp&gt;PATTERN   PCS   ETHERNET   IPUDP   OTL   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   FEC   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   EOTNPCS   QOSMETRICS   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   ODUFLEX   ODUFLEXTCM     GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM[, &lt;Channel&gt;]</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PATTERN   PCS   ETHERNET   IPUDP   OTL   ODU0   ODU1   ODU2   ODU3   OTU3   OTU3e1   OTU3e2   FEC   ODUTCM0   ODUTCM1   ODUTCM2   ODUTCM3   ODU3e1TCM   ODU3e2TCM   ODU0GMP   ODU3GMP   XCODING   EOTNPCS   QOSMETRICS   OTU4   ODU4   OPU4   ODU4TCM   ODU4GMP   ODUFLEX   ODUFLEXTCM     GFPFrame   GFPChannel   SECTION   LINE   PATH   ODU1E   ODU2E   ODU1ETCM   ODU2ETCM   OPU1E   OPU2E   HOPTCM</p> <p>Selects all errors related to the selected test and all the tests.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Rate>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:ERRor:RATE?**

<b>Response(s)</b>	<b>Rate:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the current error rate for the selected test errors and all the test errors.
<b>Example(s)</b>	FETC:DATA:TEL:ERR:RATE? ODU3 FETC:DATA:TEL:ERR:RATE? ODU3, 3
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ERRor:HISTory? FETCh[1..n]:DATA:TELEcom:ERRor:CURREnt? FETCh[1..n]:DATA:TELEcom:ERRor:SECConds? FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?

---

**:FETCh[1..n]:DATA:TELEcom:ERRor:COUNT:TOTal?**

<b>Description</b>	<p>This query returns the total count of all errors related to the test such as Ethernet, Pattern and Other. It also returns the combined status for all the tests.</p> <p>At *RST condition, this value is device dependent.</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ERRor:COUNT:TOTal? <wsp>OTL   XCODING   PCS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OTL   XCODING   PCS</p> <p>Selects all errors related to the selected test and all the tests.</p> <p>OTL, retrieves all errors related to OTL</p> <p>XCODING, retrieves all errors related to XCODING</p> <p>PCS, retrieves all errors related to PCS</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the total error count for the selected test errors and all the test errors.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ERR:COUN:TOT? OTL
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?</p> <p>FETCh[1..n]:DATA:TELEcom:ERRor:CURRent?</p> <p>FETCh[1..n]:DATA:TELEcom:ERRor:SECConds?</p> <p>FETCh[1..n]:DATA:TELEcom:ERRor:RATE?</p>

## SCPI Command Reference

### Alarms/Errors

---

:FETCh[1..n]:DATA:TELEcom:ERRor:RATE:TOTal?	
<b>Description</b>	<p>This query returns the total current rate of all errors related to the test such as Ethernet, Pattern and Other. It also returns the combined status for all the tests.</p> <p>At *RST condition, this value is device dependent.</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ERRor:RATE:TOTal? <wsp>OTL   XCODING   PCS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OTL   XCODING   PCS</p> <p>Selects all errors related to the selected test and all the tests.</p> <p>OTL, retrieves all errors related to OTL</p> <p>XCODING, retrieves all errors related to XCODING</p> <p>PCS, retrieves all errors related to PCS</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the total current error rate for the selected test errors and all the test errors.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ERR:RATE:TOT? OTL
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ERRor:HISTory? FETCh[1..n]:DATA:TELEcom:ERRor:CURRent? FETCh[1..n]:DATA:TELEcom:ERRor:SECOnds? FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE**

---

<b>Description</b>	<p>This command selects the manual type of GFP Frame error.</p> <p>At *RST condition, this value is set to GFP-CHEC-CORR Frame error.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Defect</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE &lt;wsp&gt;CORRECTABLE   UCORRECTABLE</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CORRECTABLE   UCORRECTABLE</p> <p>Selects the GFP Error type for manual injection for Frame.</p> <p>CORRECTABLE: the type of CORRECTABLE error for Frame.</p> <p>UCORRECTABLE: the type of UCORRECTABLE error for Frame.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:GFP:ERR:FRAM:MAN:TYPE UCOR SOUR:DATA:TEL:GFP:ERR:FRAM:MAN:TYPE? Returns: UCOR</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AMOUNT?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual type of GFP Frame error.</p> <p>At *RST condition, this value is set to GFP-CHEC-CORR error.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type(Errors) &gt; Mod (Manual) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of GFP error type for manual injection for Frame.</p> <p>CORRECTABLE,CORRECTABLE is selected as GFP error for manual injection for Frame.</p> <p>UCORRECTABLE,UCORRECTABLE is selected as GFP error for manual injection for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:MAN:TYPE CORR</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:MAN:TYPE?</p> <p>Returns: CORR</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AMOUnt



---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AMOUnt**

<b>Description</b>	<p>This command sets the amount for the GFP Frame error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AMOUnt &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of GFP error for Frame. Choices are 1 to 50. The default setting is 1.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AMO 40</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AMO?</p> <p>Returns: 40</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AMOUt?**

<b>Description</b>	<p>This query returns the amount for the for GFP Frame error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt;OTN BERT&gt; Test Configurator &gt; Modify Structure&gt;Client (10GbE)&gt; GFP-F&gt; Results &gt; Alarms/Errors &gt;Global Injection&gt;Layer (GFP)&gt;Type(Errors)&gt; Mode(Manual)&gt; Amount</p> <p>Navigation Path: Test Setup &gt;OTN BERT &gt; Test Configurator &gt;Modify Structure &gt; Client (1GbE)&gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection&gt; Layer (GFP) &gt;Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AMOUt?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of error is returned for Frame.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of GFP error for manual injection for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AMO 40</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AMO?</p> <p>Returns: 40</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE</p>

**:SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:INJect**

<b>Description</b>	<p>This command injects the GFP Frame error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:GFP:ERR:FRAM:INJect
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the Automated type of GFP Frame error.</p> <p>At *RST condition, this value is set to GFP-CHEC-CORR error.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:TYPE &lt;wsp&gt;CORRECTABLE   UCORRECTABLE</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CORRECTABLE   UCORRECTABLE</p> <p>Selects the GFP Error type for manual injection for Frame.</p> <p>CORRECTABLE: the type of CORRECTABLE error for Frame.</p> <p>UCORRECTABLE: the type of UCORRECTABLE error for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:TYPE CORR</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:TYPE?</p> <p>Returns: CORR</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:RATE?</p>

---

**:SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:TYPE?**

---

<b>Description</b>	<p>This query returns the Automated type of GFP Frame error.</p> <p>At *RST condition, this value is set to GFP-CHEC-CORR error.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Defect</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:TYPE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of GFP error type for manual injection for Frame.</p> <p>CORRECTABLE,CORRECTABLE is selected as GFP error for manual injection for Frame.</p> <p>UCORRECTABLE,UCORRECTABLE is selected as GFP error for manual injection for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:TYPE UCOR</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:TYPE?</p> <p>Returns: UCOR</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:RATE

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:RATE

---

<b>Description</b>	<p>This command sets the Automated rate for the GFP Frame error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:RATE <wsp> <Rate>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected GFP error for Frame.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:RATE 1.2E-09</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:RATE?</p> <p>Returns: 1.2E-09</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the Automated rate for the GFP Frame error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:RATE? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current injection rate for the selected GFP error is returned for Frame.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected GFP error for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:RATE 1.2E-09</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:RATE?</p> <p>Returns: 1.2E-09</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:TYPE

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated**

<b>Description</b>	<p>This command sets the GFP Frame automated error generation status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated GFP error injection for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT ON</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTInuous?</p>



---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated?**

---

<b>Description</b>	<p>This query returns the GFP Frame automated error generation status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated?
<b>Response Syntax</b>	<SET>
<b>Response(s)</b>	<p><b>SET:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated GFP error injection for Frame.</p> <p>1, GFP error injection is enabled for Frame.</p> <p>0, GFP error injection is disabled for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT ON</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTinuous

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:CONTInuous**

<b>Description</b>	<p>This command sets the GFP Frame automated continuous mode status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client(10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Max Rate)</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated GFP Continuous error injection for Frame.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:CONT ON SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:CONT? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated?</pre>

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:CONTInuous?**


---

<b>Description</b>	<p>This command returns the GFP Frame Automated Continuous mode status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup&gt;OTN BERT &gt;Test Configurator &gt; Modify Structure &gt; Client(10GbE)&gt;GFP-F &gt; Results &gt;Alarms/Errors &gt;Global Injection&gt;Layer (GFP) &gt;Type(Errors)&gt; Mode (Max Rate)&gt; Inject</p> <p>Navigation Path: Test Setup &gt;OTN BERT &gt; Test Configurator &gt;Modify Structure &gt; Client (1GbE)&gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection&gt;Layer (GFP) &gt;Type (Errors) &gt; Mode (Max Rate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated:CONTInuous?
<b>Response Syntax</b>	<SET>
<b>Response(s)</b>	<p><b>SET:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated GFP continuous error injection for Frame.</p> <p>1, GFP continuous error injection is enabled for Frame.</p> <p>0, GFP continuous error injection is disabled for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:CONT ON</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:AUTomated

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:GFP:ALARm:FRAMe:HISTory?

---

<b>Description</b>	<p>This query returns the history status of the GFP Frame alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:GFP:ALARm:FRAMe:HISTory? <wsp>LFD   EXM   UPM
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LFD   EXM   UPM</p> <p>Selects the history status of the GFP alarm for Frame.</p> <p>LFD</p> <p>EXM</p> <p>UPM</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of the GFP alarm for Frame.</p> <p>PRESENT, indicates that at least one GFP alarm is present for Frame.</p> <p>ABSENT, indicates that there is no GFP alarm for Frame</p> <p>INACTIVE, indicates that the test did not run yet for Frame.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ALAR:FRAM:HIST? LFD
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:GFP:ALARm:FRAMe:CURRent?

---

**:FETCh[1..n]:DATA:TELecom:GFP:ALARm:FRAMe:CURRent?**

<b>Description</b>	<p>This query returns the current status of the GFP Frame alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:GFP:ALARm:FRAMe:CURRent? <wsp>LFD   EXM   UPM
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LFD   EXM   UPM</p> <p>Selects the current status of the GFP alarm for Frame.</p> <p>LFD</p> <p>EXM</p> <p>UPM</p>
<b>Response Syntax</b>	<current>
<b>Response(s)</b>	<p><b>current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of the GFP alarm for Frame.</p> <p>PRESENT, indicates that at least one GFP alarm is present v.</p> <p>ABSENT, indicates that there is no GFP alarm for Frame</p> <p>INACTIVE, indicates that the test did not run yet for Frame.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ALAR:FRAM:CURR? LFD
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:GFP:ALARm:FRAMe:HISTory?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELeom:GFP:ALARm:FRAMe:SEConDs?

---

<b>Description</b>	<p>This query returns the number of seconds within which GFP Frame alarm occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:GFP:ALARm:FRAMe:SEConDs? <wsp>LFD   EXM   UPM
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LFD   EXM   UPM</p> <p>Selects the number of seconds within which the GFP alarm occurred for Frame.</p> <p>LFD</p> <p>EXM</p> <p>UPM</p>
<b>Response Syntax</b>	<seconds>
<b>Response(s)</b>	<p><b>seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds within which GFP alarm occurred for Frame.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ALAR:FRAM:SEC? LFD
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:GFP:ALARm:FRAMe:CURRent?

---

**:FETCh[1..n]:DATA:TELeom:GFP:ERRor:FRAMe:HISTory?**

<b>Description</b>	<p>This query returns the history status of the GFP Frame error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:GFP:ERRor:FRAMe:HISTory? <wsp>CORReactable   UCORReactable   TUCorrect
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CORReactable   UCORReactable   TUCorrect</p> <p>Selects the type of the GFP error for Frame.</p> <p>CORReactable: Correctable</p> <p>UCORReactable: UCORReactable</p> <p>TUCorrect: TUCorrect</p> <p>EUCorrect: EUCorrect</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of error Analysis for Frame.</p> <p>PRESENT, indicates that at least one GFP error is present for Frame.</p> <p>ABSENT, indicates that there is no GFP error for Frame</p> <p>INACTIVE, indicates that the test did not run yet for Frame.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ERR:FRAM:HIST? CORReactable
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:GFP:ERRor:FRAMe:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:CURRent?

<b>Description</b>	<p>This query returns the current status of GFP Frame error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:CURRent? <wsp>CORRectable   UCORRectable   TUCORrect
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CORRectable   UCORRectable   TUCORrect</p> <p>Selects the type of the GFP error for Frame.</p> <p>CORRectable: Correctable</p> <p>UCORRectable: the UCORRectable</p> <p>TUCORrect: the TUCORrect</p> <p>EUCORrect: the EUCORrect</p>
<b>Response Syntax</b>	<current>
<b>Response(s)</b>	<p><b>current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of error Analysis for Frame.</p> <p>PRESENT, indicates that at least one GFP error is present for Frame.</p> <p>ABSENT, indicates that there is no GFP error for Frame</p> <p>INACTIVE, indicates that the test did not run yet for Frame.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ERR:FRAM:CURR? CORRectable
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:HISTory?



**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:SEConDs?**

<b>Description</b>	<p>This query returns the number of seconds within which GFP Frame error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:SEConDs? <wsp>CORReactable   UCORReactable   TUCorrect
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CORReactable   UCORReactable   TUCorrect</p> <p>Selects the GFP error occurred for Frame.</p> <p>CORReactable: Correctable</p> <p>UCORReactable: UCORReactable</p> <p>TUCorrect: the TUCorrect</p> <p>EUCorrect: the EUCorrect</p>
<b>Response Syntax</b>	<seconds>
<b>Response(s)</b>	<p><b>seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns number of seconds within which GFP error occurred for Frame.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ERR:FRAM:SEC? CORReactable
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:RATE?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:GFP:ERRor:FRAMe:COUNT?

---

<b>Description</b>	<p>This query returns the count of GFP Frame error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:GFP:ERRor:FRAMe:COUNT? <wsp>CORReactable   UCORReactable   TUCorrect
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CORReactable   UCORReactable   TUCorrect</p> <p>Selects the GFP error for Frame.</p> <p>CORReactable: Correctable</p> <p>UCORReactable: UCORReactable</p> <p>TUCorrect: TUCorrect</p> <p>EUCorrect: EUCorrect</p>
<b>Response Syntax</b>	<COUNT>
<b>Response(s)</b>	<p><b>COUNT:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of error Analysis for Frame.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ERR:FRAM:COUN? CORReactable
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:GFP:ERRor:FRAMe:RATE?

---

---

**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAME:RATE?**

<b>Description</b>	<p>This query returns the current rate of GFP Frame error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Errors</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAME:RATE? &lt;wsp&gt;CORRectable   UCORrectable   TUCorrect</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CORRectable   UCORrectable   TUCorrect</p> <p>Selects the type of the GFP error for Frame.</p> <p>CORRectable: Correctable</p> <p>UCORrectable: UCORrectable</p> <p>TUCorrect: TUCorrect</p> <p>EUCorrect: EUCorrect</p>
<b>Response Syntax</b>	<p>&lt;RATE&gt;</p>
<b>Response(s)</b>	<p><b>RATE:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of error Analysis for Frame.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:GFP:ERR:FRAM:RATE? CORRectable</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAME:COUNT?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:TYPE

---

<b>Description</b>	<p>This command selects Automated type of GFP Channel error.</p> <p>At *RST condition, this value is set to tHEC-CORRectable.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Defect</p>
<b>Syntax</b>	<pre>:SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:TYPE &lt;wsp&gt;TCORrect   TUCORrect   CODERR   SUCORRPRE   SUCORRPOST   SUUNCORR</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TCORrect   TUCORrect   CODERR   SUCORRPRE   SUCORRPOST   SUUNCORR</p> <p>Selects the GFP Error type for automated injection.</p> <p>TCORrect: CORRectable TUCORrect: UCORrectable CODERR: CODERR SUCORRPRE: CORRectable SUCORRPOST: CORRectable SUUNCORR: UCORrectable ECORRECT: ECORRECT EUCORrect: EUCORRECT PFCS: PFCS</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:TYPE TCORRect SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:TYPE? Returns: TCORRect</pre>
<b>See Also</b>	<pre>SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:RATE?</pre>

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the Automated type of GFP Channel error.</p> <p>At *RST condition, this value is set to tHEC-CORReactable.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:TYPE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the error for automated injection mode.</p> <p>TCORrect,TCORrect is selected as type of CORReactable error for automated injection.</p> <p>TUCORrect,TUCORrect selected as type of UCORReactable error for automated injection.</p> <p>SUCORRRPRE,SUCORRRPRE selected as type of CORReactable error for pre automated injection for superblock.</p> <p>SUCORRRPOST,SUCORRRPOST selected as type of CORReactable error for post automated injection for superblock.</p> <p>SUUNCORR, SUUNCORR selected as type of UCORReactable error for automated injection for superblock.</p> <p>CODERR, CODERR is selected as GFP error for automated injection.</p> <p>ECORRECT, ECORRECT is selected as type of channel error</p> <p>EUCORrect, EUCORRECT is selected as type of channel error</p> <p>PFCS, PFCS is selected as type of channel error</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:TYPE TCORRect</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:TYPE?</p> <p>Returns: TCORRect</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:RATE

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:RATE**

<b>Description</b>	<p>This command sets the Automated rate for the GFP Channel error.</p> <p>At *RST condition, this value is set to 1.0E-01.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected GFP automated error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:RATE 1.2E-09</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:RATE?</p> <p>Returns: 1.2E-09</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the Automated rate for the GFP Channel error.</p> <p>At *RST condition, this value is set to 1.0E-01.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:RATE? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current injection rate for the selected GFP error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<RATE>
<b>Response(s)</b>	<p><b>RATE:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected GFP automated error for Channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:RATE 1.2E-09</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:RATE?</p> <p>Returns: 1.2E-09</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:TYPE

## SCPI Command Reference

### Alarms/Errors

---

---

#### **:SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTInuous**

---

<b>Description</b>	<p>This command generates automated continuous mode status for GFP Channel.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Max Rate)</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	<pre>:SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated GFP Continuous error injection.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:CONT ON SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:CONT? Returns: 1</pre>
<b>See Also</b>	<pre>SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated?</pre>

---



---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTInuous?**

---

<b>Description</b>	<p>This query returns the Automated Continuous mode status for GFP Channel.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup&gt;OTN BERT &gt;Test Configurator &gt; Modify Structure&gt; Client (10GbE)&gt;GFP-F &gt;Results &gt; Alarms/Errors&gt; Global Injection&gt;Layer (GFP) &gt;Type(Errors) &gt;Mode (Max Rate)</p> <p>Navigation Path: Test Setup &gt;OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE)&gt; GFP-T &gt; Results &gt; Alarms/Errors &gt;Global Injection&gt; Layer (GFP) &gt;Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTInuous?
<b>Response Syntax</b>	<SET>
<b>Response(s)</b>	<p><b>SET:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated continuous error injection for Channel.</p> <p>1, continuous error injection is enabled.</p> <p>0, continuous error injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:CONT ON</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated**

<b>Description</b>	<p>This command sets the automated error generation status for GFP Channel.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated GFP error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT ON</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTInuous?

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated?**

---

<b>Description</b>	<p>This query returns the automated error generation status for GFP Channel.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated?
<b>Response Syntax</b>	<SET>
<b>Response(s)</b>	<p><b>SET:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated error injection for Channel.</p> <p>1, error injection is enabled.</p> <p>0, error injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT ON</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTinuous

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:MANual:TYPE**

<b>Description</b>	<p>This command selects Manual injection mode for GFP Channel error.</p> <p>At *RST condition, this value is set to tHEC-CORRectable.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:MANual:TYPE &lt;wsp&gt;TCORrect   TUCORrect   CODERR   SUCORRPRE   SUCORRPOST   SUUNCORR</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TCORrect   TUCORrect   CODERR   SUCORRPRE   SUCORRPOST   SUUNCORR</p> <p>Selects the GFP Error type for automated injection.</p> <p>TCORrect: CORRectable</p> <p>TUCORrect: UCORrectable</p> <p>CODERR: CODERR</p> <p>SUCORRPRE: CORRectable</p> <p>SUCORRPOST: CORRectable</p> <p>SUUNCORR: UCORrectable</p> <p>ECORRECT: ECORRECT</p> <p>EUCORrect: EUCORRECT</p> <p>PFCS: PFCS</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:MAN:TYPE TCORRect</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:MAN:TYPE?</p> <p>Returns: TCORRect</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AMOUnt?</p>

**:SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:MANual:TYPE?**

<b>Description</b>	<p>This query returns error for manual type of GFP Channel error.</p> <p>At *RST condition, this value is set to tHEC-CORRectable.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mod (Manual) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Defect</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:MANual:TYPE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of GFP error type for manual injection for Channel.</p> <p>TCORrect, TCORrect is selected as type of CORRectable error for manual injection.</p> <p>TUCORrect, TUCORrect selected as type of UCORRectable error for manual injection.</p> <p>SUCORRRPRE, SUCORRRPRE selected as type of CORRectable error for pre manual injection.</p> <p>SUCORRPOST, SUCORRPOST selected as type of CORRectable error for post manual injection.</p> <p>SUUNCORR, SUUNCORR selected as type of UCORRectable error for manual injection.</p> <p>CODERR, CODERR is selected as GFP error for manual injection.</p> <p>ECORRECT, ECORRECT is selected as GFP error for manual injection.</p> <p>EUCORrect, EUCORrect is selected as GFP error for manual injection.</p> <p>PFCS, PFCS is selected as GFP error for manual injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:MAN:TYPE TCORReCt</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:MAN:TYPE?</p> <p>Returns: TCORReCt</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AMOunt

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AMOUnt

---

<b>Description</b>	<p>This command sets the amount for the GFP Channel manual error to be generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer(GFP) &gt; Type(Errors) &gt; Mode(Manual) &gt; Amount</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AMOUnt <wsp> <Amount>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of GFP error. Choices are 1 to 50. The default setting is 1.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AMO 50</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AMO?</p> <p>Returns: 50</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:MANual:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AMOUnt?**

<b>Description</b>	<p>This query returns the amount for the GFP Channel manual error to be generated.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AMOUnt?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<AMOUNT>
<b>Response(s)</b>	<p><b>AMOUNT:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns amount for the GFP manual error to be generated for Channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AMO 50</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AMO?</p> <p>Returns: 50</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:MANual:TYPE

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:INJect

---

<b>Description</b>	<p>This command injects the GFP Channel error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:GFP:ERR:CHAN:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:AUTomated?

---



**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:HISTory?**

<b>Description</b>	<p>This query returns the history status of the GFP Channel alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (GFP-F)/10GbE &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:HISTory? <wsp>CLOCS   LOCCS   RDI   FDI   DCI   FDI   DCI   CMF
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CLOCS   LOCCS   RDI   FDI   DCI   FDI   DCI   CMF</p> <p>Sets the history status of the GFP alarm.</p> <p>CLOCS, generation of LOC signal by setting UPI to ""0000 0001""</p> <p>LOCCS, generation of LOC signal by setting UPI to ""0000 0010""</p> <p>RDI, generation of LOC signal by setting UPI to ""0000 0101""</p> <p>FDI, generation of LOC signal by setting UPI to ""0000 0100""</p> <p>DCI, generation of LOC signal by setting UPI to ""0000 0011""</p> <p>CMF, generation of CMF signal</p>
<b>Response Syntax</b>	<History>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:HISTory?

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of the GFP alarm for Channel. PRESENT, indicates that at least one GFP alarm is present. ABSENT, indicates that there is no GFP alarm INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ALAR:CHAN:HIST? LOCS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:CURREnt?

---

**:FETCh[1..n]:DATA:TELeom:GFP:ALARm:CHANnel:CURRent?**

<b>Description</b>	<p>This query returns the current status of the GFP Channel alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:GFP:ALARm:CHANnel:CURRent? <wsp>CLOCS   LOCCS   RDI   FDI   DCI   FDI   DCI   CMF
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CLOCS   LOCCS   RDI   FDI   DCI   FDI   DCI   CMF</p> <p>Sets the history status of the GFP alarm.</p> <p>CLOCS, generation of LOC signal by setting UPI to ""0000 0001""</p> <p>LOCCS, generation of LOC signal by setting UPI to ""0000 0010""</p> <p>RDI, generation of LOC signal by setting UPI to ""0000 0101""</p> <p>FDI, generation of LOC signal by setting UPI to ""0000 0100""</p> <p>DCI, generation of LOC signal by setting UPI to ""0000 0011""</p> <p>CMF, generation of CMF signal</p>
<b>Response Syntax</b>	<Current>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:GFP:ALARm:CHANnel:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of the GFP alarm for Channel. PRESENT, indicates that at least one GFP alarm is present. ABSENT, indicates that there is no GFP alarm INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ALAR:CHAN:CURR? LOCS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:GFP:ALARm:CHANnel:HISTory?

---

**:FETCh[1..n]:DATA:TELecom:GFP:ALARm:CHANnel:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which GFP Channel alarm occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:GFP:ALARm:CHANnel:SEConds? <wsp>CLOCS   LOCCS   RDI   FDI   DCI   FDI   DCI   CMF
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CLOCS   LOCCS   RDI   FDI   DCI   FDI   DCI   CMF</p> <p>Sets the history status of the GFP alarm.</p> <p>LOCS, generation of LOC signal by setting UPI to ""0000 0001""</p> <p>LOCCS, generation of LOC signal by setting UPI to ""0000 0010""</p> <p>RDI, generation of LOC signal by setting UPI to ""0000 0101""</p> <p>FDI, generation of LOC signal by setting UPI to ""0000 0100""</p> <p>DCI, generation of LOC signal by setting UPI to ""0000 0011""</p> <p>CMF, generation of CMF signal</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds within which GFP alarm occurred for Channel.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ALAR:CHAN:SEC? LOCS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:GFP:ALARm:CHANnel:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELeom:GFP:ERRor:CHANnel:HISTory?

---

<b>Description</b>	<p>This query returns the history status of the GFP Channel error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Errors</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELeom:GFP:ERRor:CHANnel:HISTory? &lt;wsp&gt;TCORrect   SUCORR   ECORrect   SUUNCOR   GFPCODERR   PFCS</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TCORrect   SUCORR   ECORrect   SUUNCOR   GFPCODERR   PFCS</p> <p>Selects the Type of the GFP Automated Error type for automated injection.</p> <p>TCORrect: TCORrect</p> <p>SUCORR: SUCORR</p> <p>ECORrect: ECORrect</p> <p>SUUNCOR: SUUNCOR</p> <p>GFPCODERR: GFPCODERR</p> <p>PFCS: PFCS type when client data frames FCS is enabled</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>

---

**:FETCh[1..n]:DATA:TELecom:GFP:ERRor:CHANnel:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of error Analysis for Channel. PRESENT, indicates that at least one GFP error is present. ABSENT, indicates that there is no GFP error INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ERR:CHAN:HIST? TCORRect
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:GFP:ERRor:CHANnel:CURRent?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELeom:GFP:ERror:CHANnel:CURRent?

---

<b>Description</b>	<p>This query returns the current status of GFP Channel error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Errors</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELeom:GFP:ERror:CHANnel:CURRent? &lt;wsp&gt;TCORrect   SUCORR   ECORrect   SUUNCOR   GFPCODERR   PFCS</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TCORrect   SUCORR   ECORrect   SUUNCOR   GFPCODERR   PFCS</p> <p>Selects the Type of Automated Error type for automated injection.</p> <p>TCORrect: TCORrect</p> <p>SUCORR: SUCORR</p> <p>ECORrect: ECORrect</p> <p>SUUNCOR: SUUNCOR</p> <p>GFPCODERR: GFPCODERR</p> <p>PFCS: PFCS type when client data frames FCS is enabled</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>

---



**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of error Analysis for Channel. PRESENT, indicates that at least one GFP error is present. ABSENT, indicates that there is no GFP error INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ERR:CHAN:CURR? TCORRect
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:HISTory?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which GFP Channel error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:SEConds? <wsp>TCORrect   SUCORR   ECORrect   SUUNCOR   GFPCODERR   PFCS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TCORrect   SUCORR   ECORrect   SUUNCOR   GFPCODERR   PFCS</p> <p>Selects the Type pf Automated Error occurred.</p> <p>TCORrect: TCORrect</p> <p>SUCORR: UCORR</p> <p>ECORrect: ECORrect</p> <p>SUUNCOR: SUUNCOR</p> <p>GFPCODERR: GFPCODERR</p> <p>PFCS: PFCS</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns number of seconds within which GFP error occurred for Channel.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ERR:CHAN:SEC? TCORRect
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:CURRent?

---

**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:COUNT?**

<b>Description</b>	<p>This query returns the count of GFP Channel error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:COUNT? <wsp>TCORrect   SUCORR   ECORrect   SUUNCOR   GFPCODERR   PFCS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TCORrect   SUCORR   ECORrect   SUUNCOR   GFPCODERR   PFCS</p> <p>Selects the error type as GFP Automated Error type</p> <p>TCORrect: TCORrect</p> <p>SUCORR: SUCORR</p> <p>ECORrect: ECORrect</p> <p>SUUNCOR: SUUNCOR</p> <p>GFPCODERR: GFPCODERR</p> <p>PFCS: PFCS type when client data frames FCS is enabled</p>
<b>Response Syntax</b>	<COUNT>
<b>Response(s)</b>	<p><b>COUNT:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of error Analysis for Channel.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:ERR:CHAN:COUN? TCORrect
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:RATE?

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELecom:GFP:ERRor:CHANnel:RATE?**

<b>Description</b>	<p>This query returns the current rate of GFP Channel error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; Errors</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:GFP:ERRor:CHANnel:RATE? &lt;wsp&gt;TCORrect   SUCORR   ECORrect   SUUNCOR   GFPCODERR   PFCS</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TCORrect   SUCORR   ECORrect   SUUNCOR   GFPCODERR   PFCS</p> <p>Selects the error type as GFP Automated Error type</p> <p>TCORrect: TCORrect</p> <p>SUCORR: SUCORR</p> <p>ECORrect: ECORrect</p> <p>SUUNCOR: SUUNCOR</p> <p>GFPCODERR: GFPCODERR</p> <p>PFCS: PFCS type when client data frames FCS is enabled</p>
<b>Response Syntax</b>	<p>&lt;RATE&gt;</p>
<b>Response(s)</b>	<p><b>RATE:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the rate of error Analysis for Channel.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:GFP:ERR:CHAN:RATE? TCORrect</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:GFP:ERRor:CHANnel:COUNt?</p>

**:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:TYPE**

<b>Description</b>	<p>This command selects the type of GFP alarm for Frame.</p> <p>At *RST condition, this value is set to GFP-LFD.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:TYPE <wsp>LFD
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LFD</p> <p>Selects the GFP Frame alarm type for Frame.</p> <p>LFD: LFD alarm for GFP type for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:FRAM:TYPE LFD</p> <p>SOUR:DATA:TEL:GFP:ALAR:FRAM:TYPE?</p> <p>Returns: LFD</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe?

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:TYPE?**

<b>Description</b>	<p>This query returns the type of GFP alarm for Frame.</p> <p>At *RST condition, this value is set to GFP-LFD.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:TYPE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of GFP alarm for Frame.</p> <p>LFD, LFD is selected as GFP alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:FRAM:TYPE LFD</p> <p>SOUR:DATA:TEL:GFP:ALAR:FRAM:TYPE?</p> <p>Returns: LFD</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe**

<b>Description</b>	<p>This command enables or disables the GFP alarm generation for Frame.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Mode (Continuous) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Mode (Continuous) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the GFP alarm generation for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:FRAM ON</p> <p>SOUR:DATA:TEL:GFP:ALAR:FRAM?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:TYPE?

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe?**

<b>Description</b>	<p>This query returns enables or disables the GFP alarm generation for Frame.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Mode (Continuous) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Mode (Continuous) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe?
<b>Response Syntax</b>	<SET>
<b>Response(s)</b>	<p><b>SET:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the GFP alarm generation for Frame.</p> <p>1, alarm generation enabled for Frame.</p> <p>0, alarm generation disabled for Frame.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:FRAM ON</p> <p>SOUR:DATA:TEL:GFP:ALAR:FRAM?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:TYPE



**:SOURce[1..n]:DATA:TELecom:GFP:ALARm:CHANnel:TYPE**

<b>Description</b>	<p>This command selects the type of GFP alarm for channel.</p> <p>At *RST condition, this value is set to LOCS.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELecom:GFP:ALARm:CHANnel:TYPE <wsp>LOCS   LOCCS   RDI   FDI   DCI   UDCMF
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOCS   LOCCS   RDI   FDI   DCI   UDCMF</p> <p>Sets generation of LOC signal by setting UPI</p> <p>LOCS, generation of LOC signal by setting UPI to ""0000 0001""</p> <p>LOCCS, generation of LOC signal by setting UPI to ""0000 0010""</p> <p>RDI, generation of LOC signal by setting UPI to ""0000 0101""</p> <p>FDI, generation of LOC signal by setting UPI to ""0000 0100""</p> <p>DCI, generation of LOC signal by setting UPI to ""0000 0011""</p> <p>UDCMF, generation of UDCMF signal by setting CMF</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN:TYPE LOCS</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN:TYPE?</p> <p>Returns: LOCS</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:GFP:ALARm:CHANnel:PERiod?

## SCPI Command Reference

### Alarms/Errors

---

**:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:TYPE?**

<b>Description</b>	<p>This query returns the generation of LOC signal by setting UPI.</p> <p>At *RST condition, this value is set to LOCS.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:TYPE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of GFP alarm for Channel.</p> <p>LOCS, LOCS is selected as GFP alarm for Channel.</p> <p>LOCCS, LOCCS is selected as GFP alarm for Channel.</p> <p>RDI, RDI is selected as GFP alarm for Channel.</p> <p>FDI, FDI is selected as GFP alarm for Channel.</p> <p>DCI, DCI is selected as GFP alarm for Channel.</p> <p>UDCMF, UDCMF is selected as GFP alarm for Channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN:TYPE LOCS</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN:TYPE?</p> <p>Returns: LOCS</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:PERiod

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:PERiod**

---

<b>Description</b>	<p>This command sets the alarm period associated with client management frames.</p> <p>At *RST condition, this value is set to 100 ms.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Period</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:PERiod &lt;wsp&gt; &lt;Period&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Period:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the period of channel. Choices are 10 to 1200 ms.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:GFP:ALAR:CHAN:PER 100 SOUR:DATA:TEL:GFP:ALAR:CHAN:PER? Returns: 100</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:TYPE?</code>

---

## SCPI Command Reference

### Alarms/Errors

---

#### :SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:PERiod?

<b>Description</b>	<p>This query returns the alarm period associated with client management frames.</p> <p>At *RST condition, this value is set to 100 ms.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Period</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:PERiod?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. Choices are from 10 to 1200 ms.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Period>
<b>Response(s)</b>	<p><b>Period:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns alarm period associated with client management frames</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN:PER 100</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN:PER?</p> <p>Returns: 100</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:TYPE

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI**

<b>Description</b>	<p>This command sets the client management frames UPI value when User defined CMF is selected.</p> <p>At *RST condition, this value is set to 0000 0000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; UPI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; UPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI <wsp> <Upi>
<b>Parameter(s)</b>	<p><b>Upi:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the client management frames UPI value when User defined CMF is selected</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN:UPI 0</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN:UPI?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel?

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI?

---

<b>Description</b>	<p>This query returns the entering client management frames UPI value when User defined CMF is selected.</p> <p>At *RST condition, this value is set to 0000 0000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; UPI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; UPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI?
<b>Response Syntax</b>	<VALUE>
<b>Response(s)</b>	<p><b>VALUE:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the client management frames UPI value when User defined CMF is selected</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN:UPI 0</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN:UPI?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel

---

---

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel**

<b>Description</b>	<p>This command enables the generation of client management frames alarm.</p> <p>At *RST condition, this value is OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Mode (Continuous) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Mode (Continuous) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel <wsp>ON   OFF
<b>Parameter(s)</b>	<p>&lt;SET&gt;:</p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the generation of client management frames alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN ON</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel?**

<b>Description</b>	<p>This command disables the generation of client management frames alarm.</p> <p>At *RST condition, this value is OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Mode (Continuous) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (GFP) &gt; Type (Alarms) &gt; Mode (Continuous) &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel?
<b>Response Syntax</b>	<VALUE>
<b>Response(s)</b>	<p><b>VALUE:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the GFP alarm generation for Channel.</p> <p>1, alarm generation enabled for Channel.</p> <p>0, alarm generation disabled for Channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN ON</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI



---

**:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:CMF**

<b>Description</b>	<p>This command enables or disables the CMF.</p> <p>At *RST condition, this value is ON.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; CMF</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; CMF</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:CMF <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the CMF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:CONF:CMF ON</p> <p>SOUR:DATA:TEL:GFP:CONF:CMF?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFiguration:TYPE?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:CMF?**

<b>Description</b>	<p>This query returns the status of CMF.</p> <p>At *RST condition, this value is OFF.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; Alarms/Errors &gt; GFP-F &gt; CMF</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; Alarms/Errors &gt; GFP-T &gt; CMF</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:CONFig:CMF?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the CMF.</p> <p>1, returns the CMF as ON.</p> <p>0, returns the CMF as OFF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:CONF:CMF ON</p> <p>SOUR:DATA:TEL:GFP:CONF:CMF?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:CONFiguration:TYPE

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:HISTory?**

<b>Description</b>	<p>This query returns the history status of section alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:HISTory? <wsp>LOF1   SEF1   TIMS
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF1   SEF1   TIMS</p> <p>Selects the type of section alarm.</p> <p>LOF1: LOF (Loss of Frame)</p> <p>SEF1: SEF (Severely Errored Framing) and out-of-frame (OOF)</p> <p>TIMS: TIM-S (Trace Identifier Mismatch - Section)</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of section alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT ON</p> <p>FETC:DATA:TEL:SDHS:ALAR:SECT:HIST? LOF1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which section alarm occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:SEConds? <wsp>LOF1   SEF1   TIMS
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF1   SEF1   TIMS</p> <p>Selects the type of section alarm.</p> <p>LOF1: LOF (Loss of Frame)</p> <p>SEF1: SEF (Severely Errored Framing) and out-of-frame (OOF)</p> <p>TIMS: TIM-S (Trace Identifier Mismatch - Section)</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of section alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT ON</p> <p>FETC:DATA:TEL:SDHS:ALAR:SECT:SEC? LOF1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion</p>

---

**:FETCh[1..n]:DATA:TELecom:SDHSonet:ALARm:SECTion:CURRent?**

<b>Description</b>	<p>This query returns the current status of section alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:SDHSonet:ALARm:SECTion:CURRent? <wsp>LOF1   SEF1   TIMS
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF1   SEF1   TIMS</p> <p>Selects the type of section alarm.</p> <p>LOF1: LOF (Loss of Frame)</p> <p>SEF1: SEF (Severely Errored Framing) and out-of-frame (OOF)</p> <p>TIMS: TIM-S (Trace Identifier Mismatch - Section)</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of section alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that there is no alarm.</p> <p>INACTIVE, indicates that the test is not running.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT ON</p> <p>FETC:DATA:TEL:SDHS:ALAR:SECT:CURR? LOF1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELecom:SDHSonet:ALARm:SECTion:TYPE</p> <p>SOURce[1..n]:DATA:TELecom:SDHSonet:ALARm:SECTion</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:HISTory?

---

<b>Description</b>	<p>This query returns the history status of section error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:HISTory? <wsp>B1   FAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B1   FAS</p> <p>Selects the type of section error.</p> <p>B1: B1 (Error)</p> <p>FAS: Frame Alignment Signal</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of section error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:SECT:HIST? B1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:INJECT</p>

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which section error occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:SEConds? <wsp>B1   FAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B1   FAS</p> <p>Selects the type of section error.</p> <p>B1: B1 (Error)</p> <p>FAS: Frame Alignment Signal</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:SECT:SEC? B1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:INJECT</p>

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:CURRent?

<b>Description</b>	<p>This query returns the current status of section error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:CURRent? <wsp>B1   FAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B1   FAS</p> <p>Selects the type of section error.</p> <p>B1: B1 (Error)</p> <p>FAS: Frame Alignment Signal</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of section error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test is not running.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:SECT:CURR? B1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:INJECT</p>



**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:COUNT?**

<b>Description</b>	<p>This query returns the count of section error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:COUNT? <wsp>B1   FAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B1   FAS</p> <p>Selects the type of section error.</p> <p>B1: B1 (Error)</p> <p>FAS: Frame Alignment Signal</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:SECT:COUN? B1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:INJECT</p>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:RATE?**

<b>Description</b>	<p>This query returns the current rate of section error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Errors</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:RATE? &lt;wsp&gt;B1   FAS</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B1   FAS</p> <p>Selects the type of section error.</p> <p>B1: B1 (Error)</p> <p>FAS: Frame Alignment Signal</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:SECT:RATE? B1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:INJECT</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:HISTory?**

<b>Description</b>	<p>This query returns the history status of line alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:HISTory? <wsp>AIS   RDI
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   RDI</p> <p>Selects the type of line alarm.</p> <p>AIS: the AIS-L (Alarm Indication Signal - Line) which generates a ""111"" pattern for the bits 6, 7 and 8 of the K2 byte.</p> <p>RDI: the RDI-L (Remote Defect Indication - Line) which generates a ""110"" pattern for the bits 6, 7 and 8 of the K2 byte.</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of line alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE ON</p> <p>FETC:DATA:TEL:SDHS:ALAR:LINE:HIST? AIS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which line alarm occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:SEConds? <wsp>AIS   RDI
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   RDI</p> <p>Selects the type of line alarm.</p> <p>AIS: the AIS-L (Alarm Indication Signal - Line) which generates a ""111"" pattern for the bits 6, 7 and 8 of the K2 byte.</p> <p>RDI: the RDI-L (Remote Defect Indication - Line) which generates a ""110"" pattern for the bits 6, 7 and 8 of the K2 byte.</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of line alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE ON</p> <p>FETC:DATA:TEL:SDHS:ALAR:LINE:SEC? AIS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE</p>

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:CURRent?**

<b>Description</b>	<p>This query returns the current status of line alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:CURRent? <wsp>AIS   RDI
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   RDI</p> <p>Selects the type of line alarm.</p> <p>AIS: the AIS-L (Alarm Indication Signal - Line) which generates a ""111"" pattern for the bits 6, 7 and 8 of the K2 byte.</p> <p>RDI: the RDI-L (Remote Defect Indication - Line) which generates a ""110"" pattern for the bits 6, 7 and 8 of the K2 byte.</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of line alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred in the last second.</p> <p>ABSENT, indicates that there is no alarm.</p> <p>INACTIVE, indicates that the test is not running.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE ON</p> <p>FETC:DATA:TEL:SDHS:ALAR:LINE:CURR? AIS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE</p>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:HISTory?**

<b>Description</b>	<p>This query returns the history status of line error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:HISTory? <wsp>B2   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B2   REI</p> <p>Selects the type of line error.</p> <p>B2: B2 (Error)</p> <p>REI: REI-L (Remote Defect Indication Line)</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of line error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:LINE:HIST? B2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:INJECT</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which line error occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:SEConds? <wsp>B2   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B2   REI</p> <p>Selects the type of line error.</p> <p>B2: B2 (Error)</p> <p>REI: REI-L (Remote Defect Indication Line)</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:LINE:SEC? B2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMount</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:INJect</p>

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:CURRent?

<b>Description</b>	<p>This query returns the current status of line error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:CURRent? <wsp>B2   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B2   REI</p> <p>Selects the type of line error.</p> <p>B2: B2 (Error)</p> <p>REI: REI-L (Remote Defect Indication Line)</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of line error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test is not running.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:LINE:CURR? B2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:INJECT</p>



**:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:COUNT?**

<b>Description</b>	<p>This query returns the count of line error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:COUNT? <wsp>B2   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B2   REI</p> <p>Selects the type of line error.</p> <p>B2: B2 (Error)</p> <p>REI: REI-L (Remote Defect Indication Line)</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:LINE:COUN? B2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:INJect</p>

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:RATE?**

<b>Description</b>	<p>This query returns the current rate of line error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Section/Line &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; RS/MS &gt; Errors</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:RATE? &lt;wsp&gt;B2   REI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B2   REI</p> <p>Selects the type of line error.</p> <p>B2: B2 (Error)</p> <p>REI: REI-L (Remote Defect Indication Line)</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:LINE:RATE? B2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMount</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:INJect</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:HISTory?**

<b>Description</b>	<p>This query returns the history status of High Order Path (HOP) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; STS &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; AU &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:HISTory? <wsp>B3   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B3   REI</p> <p>Selects the type of HOP (High Order Path) error.</p> <p>B3: B3 (Error)</p> <p>REI: REI-P (Remote Error Indicator - Path)</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status High Order Path (HOP) error.</p> <p>PRESENT, indicates that at least one error has occurred.</p> <p>ABSENT, indicates that no error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:MAN:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:HOP:PATH:HIST? B3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOut</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJect</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which High Order Path (HOP) error occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;STS&gt;Errors</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;AU&gt;Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:SEConds? <wsp>B3   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B3   REI</p> <p>Selects the type of HOP (High Order Path) error.</p> <p>B3: B3 (Error)</p> <p>REI: REI-P (Remote Error Indicator - Path)</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:MAN:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:HOP:PATH:SEC? B3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJECT</p>

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:CURRent?**

<b>Description</b>	<p>This query returns the current status of High Order Path (HOP) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; STS &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; AU &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:CURRent? <wsp>B3   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B3   REI</p> <p>Selects the type of HOP (High Order Path) error.</p> <p>B3: B3 (Error)</p> <p>REI: REI-P (Remote Error Indicator - Path)</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of High Order Path (HOP) error.</p> <p>PRESENT, indicates that at least one error has occurred in the last second.</p> <p>ABSENT, indicates that there is no error.</p> <p>INACTIVE, indicates that the test is not running.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:MAN:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:HOP:PATH:CURR? B3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOut</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJect</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:COUNt?

---

<b>Description</b>	<p>This query returns the count of High Order Path (HOP) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; STS &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; AU &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:COUNt? <wsp>B3   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B3   REI</p> <p>Selects the type of HOP (High Order Path) error.</p> <p>B3: B3 (Error)</p> <p>REI: REI-P (Remote Error Indicator - Path)</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:MAN:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:HOP:PATH:COUN? B3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOut</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJect</p>

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:RATE?**

<b>Description</b>	<p>This query returns the current rate of High Order Path (HOP) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; STS &gt; Errors</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; AU &gt; Errors</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:RATE? <wsp>B3   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B3   REI</p> <p>Selects the type of HOP (High Order Path) error.</p> <p>B3: B3 (Error)</p> <p>REI: REI-P (Remote Error Indicator - Path)</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:MAN:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:HOP:PATH:RATE? B3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHS:ERRor:HOP:PATH:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHS:ERRor:HOP:PATH:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHS:ERRor:HOP:PATH:INJECT</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:HISTory?

---

<b>Description</b>	<p>This query returns the history status of High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; STS &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; AU &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:HISTory? <wsp>AIS   LOP   LOM   RDI   TIM   PLM   UNEQ   PDI   EPSD1   EPCD1   EPPD1
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   LOP   LOM   RDI   TIM   PLM   UNEQ   PDI   EPSD1   EPCD1   EPPD1</p> <p>Selects the type of HOP (High Order Path) alarm.</p> <p>AIS: AIS-P (Alarm Indication Signal - Path)</p> <p>LOP: LOP-P (Loss of Pointer - Path)</p> <p>LOM: LOM (Loss of Multiframe)</p> <p>RDI: RDI-P (Remote Defect Indication - Path)</p> <p>TIM: TIM-P (Trace Identifier Mismatch - Path)</p> <p>PLM: PLM-P (Payload Label Mismatch - Path)</p> <p>UNEQ: UNEQ-P (Unequipped - Path)</p> <p>PDI: PDI-P (Payload Defect Indication - Path)</p> <p>EPSD1: ERDI-PSD (Enhanced RDI - Path Server Defect)</p> <p>EPCD1: ERDI-PCD (Enhanced RDI - Path Connectivity Defect)</p> <p>EPPD1: ERDI-PPD (Enhanced RDI - Path Payload Defect)</p>
<b>Response Syntax</b>	<History>

---



**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of High Order Path (HOP) alarm. PRESENT, indicates that at least one alarm has occurred. ABSENT, indicates that no alarm occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:TYPE AIS SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH ON FETC:DATA:TEL:SDHS:ALAR:HOP:PATH:HIST? AIS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which High Order Path (HOP) alarm occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;STS&gt;Alarms</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;AU&gt;Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:SEConds? <wsp>AIS   LOP   LOM   RDI   TIM   PLM   UNEQ   PDI   EPSD1   EPCD1   EPPD1
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   LOP   LOM   RDI   TIM   PLM   UNEQ   PDI   EPSD1   EPCD1   EPPD1</p> <p>Selects the type of HOP (High Order Path) alarm.</p> <p>AIS: AIS-P (Alarm Indication Signal - Path)</p> <p>LOP: LOP-P (Loss of Pointer - Path)</p> <p>LOM: LOM (Loss of Multiframe)</p> <p>RDI: RDI-P (Remote Defect Indication - Path)</p> <p>TIM: TIM-P (Trace Identifier Mismatch - Path)</p> <p>PLM: PLM-P (Payload Label Mismatch - Path)</p> <p>UNEQ: UNEQ-P (Unequipped - Path)</p> <p>PDI: PDI-P (Payload Defect Indication - Path)</p> <p>EPSD1: ERDI-PSD (Enhanced RDI - Path Server Defect)</p> <p>EPCD1: ERDI-PCD (Enhanced RDI - Path Connectivity Defect)</p> <p>EPPD1: ERDI-PPD (Enhanced RDI - Path Payload Defect)</p>
<b>Response Syntax</b>	<Seconds>

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:SEConds?**

<b>Response(s)</b>	<b>Seconds:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of High Order Path (HOP) alarm.
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:TYPE AIS SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH ON FETC:DATA:TEL:SDHS:ALAR:HOP:PATH:SEC? AIS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:CURRent?

---

<b>Description</b>	<p>This query returns the current status of High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; STS &gt; Alarms</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; AU &gt; Alarms</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:CURRent? <wsp>AIS   LOP   LOM   RDI   TIM   PLM   UNEQ   PDI   EPSD1   EPCD1   EPPD1
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   LOP   LOM   RDI   TIM   PLM   UNEQ   PDI   EPSD1   EPCD1   EPPD1</p> <p>Selects the type of HOP (High Order Path) alarm.</p> <p>AIS: AIS-P (Alarm Indication Signal - Path)</p> <p>LOP: LOP-P (Loss of Pointer - Path)</p> <p>LOM: LOM (Loss of Multiframe)</p> <p>RDI: RDI-P (Remote Defect Indication - Path)</p> <p>TIM: TIM-P (Trace Identifier Mismatch - Path)</p> <p>PLM: PLM-P (Payload Label Mismatch - Path)</p> <p>UNEQ: UNEQ-P (Unequipped - Path)</p> <p>PDI: PDI-P (Payload Defect Indication - Path)</p> <p>EPSD1: ERDI-PSD (Enhanced RDI - Path Server Defect)</p> <p>EPCD1: ERDI-PCD (Enhanced RDI - Path Connectivity Defect)</p> <p>EPPD1: ERDI-PPD (Enhanced RDI - Path Payload Defect)</p>
<b>Response Syntax</b>	<Current>

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of High Order Path (HOP) alarm. PRESENT, indicates that at least one alarm has occurred in the last second. ABSENT, indicates that there is no alarm. INACTIVE, indicates that the test is not running.
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:TYPE AIS SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH ON FETC:DATA:TEL:SDHS:ALAR:HOP:PATH:CURR? AIS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:TYPE**

<b>Description</b>	<p>This command selects the type of section/RS alarm.</p> <p>At *RST condition, this value is set to LOF1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Defect</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:TYPE &lt;wsp&gt;LOF1   SEF1</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF1   SEF1</p> <p>Selects the type of section alarm.</p> <p>LOF1: LOF (Loss of Frame)</p> <p>SEF1: SEF (Severely Errored Framing) and out-of-frame (OOF)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:TYPE?</p> <p>Returns: LOF1</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion?</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:TYPE?**

---

<b>Description</b>	<p>This query returns the type of section alarm.</p> <p>At *RST condition, this value is set to LOF1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of section alarm.</p> <p>LOF1, Loss of Frame (LOF) is selected as section alarm.</p> <p>SEF1, Severely Errored Framing (SEF) is selected as section alarm and out-of-frame (OOF).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:TYPE?</p> <p>Returns: LOF1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion**

<b>Description</b>	<p>This command enables or disables the section alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section )&gt; (Type)Alarms&gt;Inject</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS )&gt; (Type)Alarms&gt;Inject</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the section alarm generation.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:TYPE?</p>



---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion?**

---

<b>Description</b>	<p>This query returns the status of section alarm.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of section alarm generation.</p> <p>0 - alarm generation enabled</p> <p>1 - alarm generation disabled</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE	
<b>Description</b>	<p>This command sets the manual error type of section error.</p> <p>At *RST condition, this value is set to B1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Errors&gt;Mode(Manual)&gt;Defect</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Errors&gt;Mode(Manual)&gt;Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE <wsp>B1   FAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B1   FAS</p> <p>Selects the type of Section error.</p> <p>B1: B1</p> <p>FAS: Frame Alignment Signal</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE?</p> <p>Returns: B1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMOut?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual error type of section error.</p> <p>At *RST condition, this value is set to B1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type)Errors &gt; Mode(Manual) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type)Errors &gt; Mode(Manual) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of section error.</p> <p>B1, B1 is selected as section error.</p> <p>FAS, Frame Alignment Signal (FAS) is selected as section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE?</p> <p>Returns: B1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMount

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMOunt

---

<b>Description</b>	<p>This command sets the amount of section error to inject into the instrument.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Errors&gt;Mode(Manual)&gt;Amount</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Errors&gt;Mode(Manual)&gt;Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMOunt <wsp> <Amount>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of section error.</p> <p>Choices are 1 through 50. The default setting is 1.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMOUnt?**

<b>Description</b>	<p>This query returns the amount of section error to injected into the instrument.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Errors &gt; Mode(Manual) &gt; Amount</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Errors &gt; Mode(Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AMOUnt?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of error is returned.</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:MAN:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:MANual:TYPE

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:INJect**

<b>Description</b>	<p>This command injects the type of section error.</p> <p>This command is an event and has no associated *RST condition or query form.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; Type (Errors) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; Type (Errors) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ERR:SECT:AMO 15 SOUR:DATA:TEL:SDHS:ERR:SECT:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHS:ERRor:SECTion:MANual:TYPE SOURce[1..n]:DATA:TELEcom:SDHS:ERRor:SECTion:AMOut

---

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERROR:SECTION:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the type of Section error for automated injection.</p> <p>At *RST condition, this value is set to B1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Errors &gt; Mode(Rate/MaxRate) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Errors &gt; Mode(Rate/MaxRate) &gt; Defect</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERROR:SECTION:AUTomated:TYPE <wsp>B1   FAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B1   FAS</p> <p>Selects the type of automated section error.</p> <p>B1: B1</p> <p>FAS: Frame Alignment Signal</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:TYPE?</p> <p>Returns: B1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERROR:SECTION:AUTomated?

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the type of Section error for automated injection.</p> <p>At *RST condition, this value is set to B1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Errors&gt;Mode(Rate/MaxRate)&gt;Defect</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Errors&gt;Mode(Rate/MaxRate)&gt;Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Section error for the automated injection.</p> <p>B1, B1 is selected as section error.</p> <p>FAS, FAS is selected as section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:TYPE?</p> <p>Returns: B1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated



---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:RATE**

---

<b>Description</b>	<p>This command sets the injection rate for the selected Section error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type)Errors &gt; Mode(Rate)</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type)Errors &gt; Mode(Rate)</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected Section error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:RATE 1.0E-10 SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:RATE? Returns: 1.0E-10</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:TYPE?</code>

---

## SCPI Command Reference

### Alarms/Errors

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:RATE?**

---

<b>Description</b>	<p>This query returns the injection rate for the selected Section error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Errors&gt;Mode(Rate)</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Errors&gt;Mode(Rate)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:RATE?[\n&lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the injected rate is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected Section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:RATE?</p> <p>Returns: 1.0E-10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:TYPE</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated**

<b>Description</b>	<p>This command enables or disables the selected automated Section error at the rate specified or continuously.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type)Errors &gt; Mode(Rate/Max Rate) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type)Errors &gt; Mode(Rate/Max Rate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Section error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:RATE?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated?

---

<b>Description</b>	<p>This query returns the status of automated Section error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type)Errors &gt; Mode (Rate/Max Rate) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type)Errors &gt; Mode (Rate/Max Rate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated Section error injection.</p> <p>1 - automated error generation enabled</p> <p>0 - automated error generation disabled</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:RATE

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:CONTInuous**

<b>Description</b>	<p>This command enables or disables the continuous rate of automated Section error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; Type (Errors) &gt; Max Rate</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; Type (Errors) &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:CONTInuous <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the rate of automated Section error injection as continuously.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:CONT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:CONTInuous?**

<b>Description</b>	<p>This query returns the status of continuous rate of automated Section error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; Type (Errors) &gt; Max Rate</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; Type (Errors) &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of continuous rate of automated Section error injection.</p> <p>1 - continuous mode status enabled</p> <p>0 -continuous mode status disabled</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:CONT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:AUTomated</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE**

---

<b>Description</b>	<p>This command selects the Burst error type for the section error.</p> <p>At *RST condition, this value is set to B1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE <wsp>B1   FAS
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B1   FAS</p> <p>Selects the Burst type for the section error.</p> <p>B1: B1 (ERROR)</p> <p>FAS: Frame Alignment Signal</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURSt:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURSt:TYPE?</p> <p>Returns: B1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE?**

<b>Description</b>	<p>This query returns the Burst error type for the section error.</p> <p>At *RST condition, this value is set to B1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)&gt;Defect</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)&gt;Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Burst type for the section error.</p> <p>B1, B1 (ERROR) is selected as section error.</p> <p>FAS, Frame Alignment Signal (FAS) is selected as section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURSt:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURSt:TYPE?</p> <p>Returns: B1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE



---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE**

---

<b>Description</b>	<p>This command selects the Burst mode for the section error.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE &lt;wsp&gt;SINGLE   REPEAT</p>
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SINGLE   REPEAT</p> <p>Selects the Burst mode for the section error.</p> <p>SINGLE: Single</p> <p>REPEAT: Repeat</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURSt:MODE SINGLE</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURSt:MODE?</p> <p>Returns: SINGLE</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE?**

<b>Description</b>	<p>This query returns the Burst mode for the section error.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Burst mode for the section error.</p> <p>SINGLE, Single is selected as Burst mode.</p> <p>REPEAT, Repeat is selected as Burst mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS:MODE SINGLE</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS:MODE?</p> <p>Returns: SINGLE</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:DURation**

<b>Description</b>	<p>This command sets the burst duration for the section error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:DURation <wsp> <Duration>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Duration:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the duration for the section error.</p> <p>Choices are 1 to 14400000 FRAMES or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:DURation?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:DURation?**

<b>Description</b>	<p>This query returns the burst duration for the section error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)&gt;Duration</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)&gt;Duration</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:DURation? [ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current duration is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Duration&gt;</p>
<b>Response(s)</b>	<p><b>Duration:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the duration for the section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:PERiod**

<b>Description</b>	<p>This command sets the burst period for the section error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type)Errors &gt; Mode(Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type)Errors &gt; Mode(Burst Repeat) &gt; Period</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:PERiod <wsp> <Period>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Period:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the period for the section error.</p> <p>Choices are 1 through 14400000 FRAMES or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURSt:PER 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURSt:PER?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:PERiod?**

<b>Description</b>	<p>This query returns the burst period for the section error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Errors &gt;Mode(Burst Repeat)&gt;Period</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Errors &gt;Mode(Burst Repeat)&gt;Period</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:PERiod? [ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current period is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Period&gt;</p>
<b>Response(s)</b>	<p><b>Period:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the period for the section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS:PER 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS:PER?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE</p>

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt**

<b>Description</b>	<p>This command enables or disables the Burst generation for the section/RS error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Burst for the section error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE?</p> <p>SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:DURation?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt?**

<b>Description</b>	<p>This query returns the status of Burst generation for the section/RS error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)&gt;Inject</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)&gt;Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Burst for the section error.</p> <p>1 - burst generation is enabled.</p> <p>0 - burst generation is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS:TYPE B1</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS:BURS:DUR 10</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:SECT:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:DURation</p>



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE**

<b>Description</b>	<p>This command selects the Burst alarm type for the Section alarm.</p> <p>At *RST condition, this value is set to LOF1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE <wsp>LOF1   SEF1
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOF1   SEF1</p> <p>Selects the Burst type for the Regenerator Section (RS) alarm.</p> <p>LOF1: Loss of Frame (LOF1)</p> <p>SEF1: Out of Frame (SEF1)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURSt:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURSt:TYPE?</p> <p>Returns: LOF1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE?

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE?**

<b>Description</b>	<p>This query returns the Burst alarm type for the Section alarm.</p> <p>At *RST condition, this value is set to LOF1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Burst type for the Section (RS) alarm.</p> <p>LOF1, Loss of Frame (LOF1) is selected as Section (RS) alarm.</p> <p>SEF1, Out of Frame (SEF1) is selected as Section (RS) alarm and out-of-frame (OOF).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:TYPE?</p> <p>Returns: LOF1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE**

<b>Description</b>	<p>This command selects the Burst mode for the Section alarm.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Mode (Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Mode (Burst Single/Burst Repeat)</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE &lt;wsp&gt;SINGLE   REPEAT</pre>
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SINGLE   REPEAT</p> <p>Selects the Burst mode for the Regenerator Section (RS) alarm.</p> <p>SINGLE: Single</p> <p>REPEAT: Repeat</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURSt:TYPE LOF1 SOUR:DATA:TEL:SDHS:ALAR:SECT:BURSt:MODE SINGLE SOUR:DATA:TEL:SDHS:ALAR:SECT:BURSt:MODE? Returns: SINGLE</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE?</pre>

## SCPI Command Reference

### Alarms/Errors

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE?

<b>Description</b>	<p>This query returns the Burst mode for the Section alarm.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Alarms&gt;Mode(Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Alarms &gt;Mode(Burst Single/Burst Repeat)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Burst mode for the Section (RS) alarm.</p> <p>SINGLE, Single is selected as Burst mode.</p> <p>REPEAT, Repeat is selected as Burst mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:MODE SINGLE</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:MODE?</p> <p>Returns: SINGLE</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:DURation**

<b>Description</b>	<p>This command sets the burst duration for the Section alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:DURation <wsp> <Duration>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Duration:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element. The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the duration for the Regenerator Section (RS) alarm.</p> <p>Choices are 1 through 14400000 or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:DURation?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:DURation?**

<b>Description</b>	<p>This query returns the burst duration for the Section alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Section)&gt;(Type)Alarms &gt;Mode(Burst Single/Burst Repeat)&gt;Duration</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(RS)&gt;(Type)Alarms &gt;Mode(Burst Single/Burst Repeat)&gt;Duration</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:DURation? [ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current duration is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Duration&gt;</p>
<b>Response(s)</b>	<p><b>Duration:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the duration for the Section (RS) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:DURation</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:PERiod**

<b>Description</b>	<p>This command sets the burst period for the Section alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Mode(Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Mode(Burst Repeat) &gt; Period</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:PERiod <wsp> <Period>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Period:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the period for the Regenerator Section (RS) alarm.</p> <p>Choices are 1 through 14400000 or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:PER 15</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:PER?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:DURation</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:PERiod?**

<b>Description</b>	<p>This query returns the burst period for the Section alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Mode(Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Mode(Burst Repeat) &gt; Period</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt:PERiod? [ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current period is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Period&gt;</p>
<b>Response(s)</b>	<p><b>Period:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the period for the Section (RS) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:PER 15</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:PER?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:DURation</p>



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt**

<b>Description</b>	<p>This command enables or disables the Burst generation for the Section alarm.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Burst for the Section alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:DURation</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt?

---

<b>Description</b>	<p>This query returns the status of Burst generation for the Section alarm.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Section) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(RS) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:SECTion:BURSt?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status for the Section (RS) alarm.</p> <p>1 - alarm burst generation is enabled.</p> <p>0 - alarm burst generation is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS:TYPE LOF1</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:SECT:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt:DURation</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:SECTion:BURSt</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:TYPE**

---

**Description**

This command selects the type of line alarm.

At \*RST condition, this value is set to AIS.

Navigation Path: Test Setup > OTN SONET/SDH BERT OR SONET/SDH BERT > Test Configurator > Results > Alarms/Errors > SONET > Layer (Line) > (Type) Alarms > Defect

Navigation Path: Test Setup > OTN SONET/SDH BERT OR SONET/SDH BERT > Test Configurator > Results > Alarms/Errors > SDH > Layer (MS) > (Type) Alarms > Defect

**Syntax**

:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:TYPE <wsp>AIS | RDI

**Parameter(s)****Alarm:**

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: AIS | RDI

Selects the type of line alarm.

AIS: the AIS-L (Alarm Indication Signal - Line) which generates a ""111"" pattern for the bits 6, 7 and 8 of the K2 byte.

RDI: the RDI-L (Remote Defect Indication - Line) which generates a ""110"" pattern for the bits 6, 7 and 8 of the K2 byte.

**Example(s)**

SOUR:DATA:TEL:SDHS:ALAR:LINE:TYPE AIS

SOUR:DATA:TEL:SDHS:ALAR:LINE:TYPE?

Returns: AIS

**See Also**

SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:TYPE?**

<b>Description</b>	<p>This query returns the type of line alarm.</p> <p>At *RST condition, this value is set to AIS.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer (Line) &gt;(Type) Alarms &gt;Defect</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer (MS) &gt;(Type) Alarms &gt;Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of line alarm.</p> <p>AIS, Alarm Indication Signal - Line (AIS-L) is selected as line alarm.</p> <p>RDI, Remote Defect Indication - Line (RDI-L) is selected as line alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE:TYPE?</p> <p>Returns: AIS</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE?

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE**

---

<b>Description</b>	<p>This command enables or disables the line alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer (Line) &gt; (Type) Alarms &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer (MS) &gt; (Type) Alarms &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables and disables the line alarm generation.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ALAR:LINE:TYPE AIS SOUR:DATA:TEL:SDHS:ALAR:LINE ON SOUR:DATA:TEL:SDHS:ALAR:LINE? Returns: 1</pre>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE?**

<b>Description</b>	<p>This query returns the status of line alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer (Line) &gt;(Type) Alarms &gt;Inject</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer (MS) &gt;(Type) Alarms &gt;Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of line alarm generation.</p> <p>0 - alarm generation disabled.</p> <p>1- alarm generation enabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:TYPE

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE**

<b>Description</b>	<p>This command sets the manual error type of line error.</p> <p>At *RST condition, this value is set to B2.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Manual) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Manual) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE <wsp>B2   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B2   REI</p> <p>Selects the type of Line error.</p> <p>B2: B2</p> <p>REI: REI-L (Remote Error Indication - Line)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE?</p> <p>Returns: B2</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMOUnt?

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual error type of line error.</p> <p>At *RST condition, this value is set to B2.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Line)&gt;(Type)Errors &gt; Mode(Manual) &gt; Defect</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(MS)&gt;(Type)Errors &gt; Mode(Manual) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of line error.</p> <p>B2, B2 is selected as line error.</p> <p>REI, Remote Error Indication - Line (REI-L) is selected as line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE?</p> <p>Returns: B2</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMOUNT



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMOunt**

<b>Description</b>	<p>This command sets the amount of line error to inject into the instrument.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Manual) &gt; Amount</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMOunt <wsp> <Amount>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of line error.</p> <p>Choices are 1 through 50. The default setting is 1.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMOUt?

---

<b>Description</b>	<p>This query returns the amount of line error injected into the instrument.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Line)&gt;(Type)Errors &gt; Mode(Manual) &gt; Amount</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(MS)&gt;(Type)Errors &gt; Mode(Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMOUt?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:INJect**

<b>Description</b>	<p>This command injects the type of line error.</p> <p>This command is an event and has no associated *RST condition or query form.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; Type (Errors) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; Type (Errors) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:INJect
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:MAN:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:INJect</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AMOUNT</p>

---

## SCPI Command Reference

### Alarms/Errors

---

:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE	
<b>Description</b>	<p>This command selects the type of Line error for automated injection.</p> <p>At *RST condition, this value is set to B2.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Line)&gt; (Type)Errors&gt; Mode(Rate/MaxRate) &gt;Defect</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(MS)&gt; (Type)Errors&gt; Mode(Rate/MaxRate) &gt;Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE <wsp>B2   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B2   REI</p> <p>Selects the type of Line error for automated injection.</p> <p>B2: B2</p> <p>REI: REI-L (Remote Error Indication - Line)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:TYPE?</p> <p>Returns: B2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:RATE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated?</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the type of Line error for automated injection.</p> <p>At *RST condition, this value is set to B2.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Rate/MaxRate) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Rate/MaxRate) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of Line error for the automated injection.</p> <p>B2, B2 is selected as line error.</p> <p>REI, Remote Error Indication - Line (REI-L) is selected as line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:TYPE?</p> <p>Returns: B2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:RATE**

<b>Description</b>	<p>This command sets the injection rate for the selected Line error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Rate)</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Rate)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected Line error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:RATE?</p> <p>Returns: 1.0E-10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated?</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the injection rate for the selected Line error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Rate)</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Rate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:RATE? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the injected rate is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected Line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:RATE?</p> <p>Returns: 1.0E-10</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated

---

<b>Description</b>	<p>This command enables or disables the selected automated Line error at the rate specified or continuously.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Rate/MaxRate) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated Line error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:RATE?</p>

---



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated?**

<b>Description</b>	<p>This query returns the status of automated Line error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Rate/MaxRate) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated Line error injection.</p> <p>0 - automated error injection disabled.</p> <p>1 - automated error injection enabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:RATE</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:CONTInuous**

<b>Description</b>	<p>This command enables or disables the continuously rate of automated Line error injection. At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Line)&gt; (Type)Errors&gt; Max Rate</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(MS)&gt; (Type)Errors&gt; Max Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the rate of automated Line error injection as continuously.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:CONT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:CONTInuous?**

<b>Description</b>	<p>This query returns the status of continuously rate of automated Line error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Max Rate</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of continuously rate of automated Line error injection.</p> <p>0 - automated error continuous mode disabled.</p> <p>1 - automated error continuous mode enabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:CONT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:AUTomated?</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE

---

<b>Description</b>	<p>This command selects the Burst error type for the line error.</p> <p>At *RST condition, this value is set to B2.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE <wsp>B2   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B2   REI</p> <p>Selects the Burst type for the line error.</p> <p>B2: B2 (Error)</p> <p>REI: Remote Error Indication - Line (REI-L)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:TYPE?</p> <p>Returns: B2</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:TYPE?

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE?**

---

<b>Description</b>	<p>This query returns the Burst error type for the line error.</p> <p>At *RST condition, this value is set to B2.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Burst type for the line error.</p> <p>B2, B2 (Error) is selected as line error.</p> <p>REI, Remote Error Indication - Line (REI) is selected as line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURSt:TYPE B2</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURSt:TYPE?</p> <p>Returns: B2</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE

---

<b>Description</b>	<p>This command selects the Burst mode for the line error.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Line)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(MS)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE <wsp>SINGLE   REPEAT
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SINGLE   REPEAT</p> <p>Selects the Burst mode for the line error.</p> <p>SINGLE: Single</p> <p>REPEAT: Repeat</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:MODE SINGLE</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:MODE?</p> <p>Returns: SINGLE</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE?**

<b>Description</b>	<p>This query returns the Burst mode for the line error.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Burst mode for the line error.</p> <p>SINGLE, Single is selected as Burst mode.</p> <p>REPEAT, Repeat is selected as Burst mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:MODE?</p> <p>Returns: SINGLE</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:DURation**

<b>Description</b>	<p>This command sets the burst duration for the Line error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:DURation &lt;wsp&gt; &lt;Duration&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Duration:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the duration for the line error.</p> <p>Choices are 1 through 14400000 or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:DURation?</p>



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:DURation?**

<b>Description</b>	<p>This query returns the burst duration for the Line error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:DURation? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current duration is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Duration>
<b>Response(s)</b>	<p><b>Duration:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the duration for the line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE?</p>

## SCPI Command Reference

### Alarms/Errors

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:PERiod

<b>Description</b>	<p>This command sets the burst period for the line error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Burst Repeat) &gt; Period</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:PERiod <wsp> <Period>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Period:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the period for the line error.</p> <p>Choices are 1 through 14400000 or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:PER 15 SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:PER? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE</pre>

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:PERiod?**

<b>Description</b>	<p>This query returns the burst period for the line error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Burst Repeat) &gt; Period</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:PERiod? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current period is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Period>
<b>Response(s)</b>	<p><b>Period:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the period for the line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:MODE REP</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:PER 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:PER?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE?</p> <p>SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE?</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt**

<b>Description</b>	<p>This command enables or disables the Burst for the line error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Line)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)&gt;Inject</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(MS)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)&gt;Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Burst for the line error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:DUR 10</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:DURation?</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt?**

<b>Description</b>	<p>This query returns the status of Burst for the line error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Burst for the line error.</p> <p>1 - line burst error is enabled.</p> <p>0 - line burst error is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS:DUR 10</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:LINE:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:DURation</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:TYPE**

<b>Description</b>	<p>This command selects the Burst alarm type for the Line alarm.</p> <p>At *RST condition, this value is set to AIS.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:TYPE <wsp>AIS   RDI
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   RDI</p> <p>Selects the Burst alarm type for the Line alarm.</p> <p>AIS: Multiplex Section - Alarm Indication Signal (MSAIS)</p> <p>RDI: Multiplex Section - Remote Defect Indication (MSRDI)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:TYPE?</p> <p>Returns: AIS</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:TYPE?**

<b>Description</b>	<p>This query returns the Burst alarm type for the Line alarm.</p> <p>At *RST condition, this value is set to AIS.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type) Alarms &gt; Mode (Burst Single/Burst Repeat) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type) Alarms &gt; Mode (Burst Single/Burst Repeat) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:TYPE?
<b>Response Syntax</b>	<alarm>
<b>Response(s)</b>	<p><b>alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Selects the Burst alarm type for the Multiplex Section (MS) alarm.</p> <p>AIS, Multiplex Section - Alarm Indication Signal (AIS) is selected as Multiplex Section (MS) alarm.</p> <p>RDI, Multiplex Section - Remote Defect Indication (RDI) is selected as Multiplex Section (MS) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:TYPE?</p> <p>Returns: AIS</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:MODE

---

<b>Description</b>	<p>This command selects the burst mode for the Line alarm.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(Line)&gt;(Type)Alarms &gt;Mode(Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(MS)&gt;(Type)Alarms&gt;Mode(Burst Single/Burst Repeat)</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:MODE &lt;wsp&gt;SINGLE   REPEAT</pre>
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SINGLE   REPEAT</p> <p>Selects the Burst mode for the Line alarm.</p> <p>SINGLE: Single</p> <p>REPEAT: Repeat</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:MODE SINGLE SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:MODE? Returns: SINGLE</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE? SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE?</pre>

---



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:MODE?**

<b>Description</b>	<p>This query returns the burst mode for the Line alarm.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type) Alarms &gt; Mode (Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type) Alarms &gt; Mode (Burst Single/Burst Repeat)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Burst mode for the Multiplex Section (MS) alarm.</p> <p>SINGLE, Single is selected as Burst mode.</p> <p>REPEAT, Repeat is selected as Burst mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:MODE SINGLE</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:MODE?</p> <p>Returns: SINGLE</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE</p>

---

## SCPI Command Reference

### Alarms/Errors

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:DURation

<b>Description</b>	<p>This command sets the burst duration for the Line alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:DURation &lt;wsp&gt; &lt;Duration&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Duration:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the burst duration for the Line alarm.</p> <p>Choices are 1 through 14400000 or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:DUR 15 SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:DUR? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE? SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE? SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:DURation?</pre>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:DURation?**

<b>Description</b>	<p>This query returns the burst duration for the Line alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:DURation? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current duration is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Duration>
<b>Response(s)</b>	<p><b>Duration:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the burst duration for the Multiplex Section (MS) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:DURation</p>

## SCPI Command Reference

### Alarms/Errors

---

#### :SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:PERiod

<b>Description</b>	<p>This command sets the burst period for the Line alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type) Alarms &gt; Mode(Burst Repeat) &gt; Duration</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type) Alarms &gt; Mode(Burst Repeat) &gt; Duration</p>
<b>Syntax</b>	<pre>:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:PERiod &lt;wsp&gt; &lt;Period&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Period:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the burstperiod for the Line alarm.</p> <p>Choices are 1 through 14400000 or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:PER 15 SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:PER? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE? SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE? SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:PERiod?</pre>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:PERiod?**

<b>Description</b>	<p>This query returns the burst period for the Line alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type)Alarms &gt; Mode(Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type)Alarms &gt; Mode(Burst Repeat) &gt; Period</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt:PERiod?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current period is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Period>
<b>Response(s)</b>	<p><b>Period:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the burst period for the Multiplex Section (MS) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:PER 15</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS:PER?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:MODE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:PERiod</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt

---

<b>Description</b>	<p>This command enables or disables the Burst generation for the Line alarm.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Period</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARm:LINE:BURSt <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Burst generation for the Line alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt:TYPE</p> <p>SOURCE[1..n]:DATA:TELEcom:SDHSONet:ERRor:LINE:BURSt?</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt?**

<b>Description</b>	<p>This query returns the status of Burst generation for the Line alarm.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(Line) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(MS) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:LINE:BURSt?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of burst generation for the Multiplex Section (MS) alarm.</p> <p>1 - line burst alarm is enabled.</p> <p>0 - line burst alarm is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:LINE:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:LINE:BURSt</p>

---

## SCPI Command Reference

### *Alarms/Errors*

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE**

#### **Description**

This command selects the type of High Order Path (HOP) alarm.

At \*RST condition, this value is set to AIS.

Navigation Path: Test Setup >OTN SONET/SDH BERT OR SONET/SDH BERT> Test Configurator >Results>Alarms/Errors>SONET>Layer (STS Path) >(Type) Alarms >Defect

Navigation Path: Test Setup >OTN SONET/SDH BERT OR SONET/SDH BERT> Test Configurator >Results>Alarms/Errors>SDH>Layer (AU Path) >(Type) Alarms >Defect

#### **Syntax**

:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE <wsp>AIS | RDI | EPSD1 | EPCD1 | EPPD1 | LOP | PDI | UNEQP1

---



---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE**

<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   RDI   EPSD1   EPCD1   EPPD1   LOP   PDI   UNEQP1</p> <p>Selects the type of HOP (High Order Path) alarm.</p> <p>AIS: the AIS-P (Alarm Indication Signal - Path) which generates all-ones pattern for the path and payload.</p> <p>RDI: the RDI-P (Remote Defect Indication - Path) which generates a ""100"" pattern for bits 5, 6 and 7 of the G1 byte.</p> <p>EPSD1: the ERDI-PSD (Enhanced RDI - Path Server Defect) which generates a ""101"" pattern for bits 5, 6 and 7 of the G1 byte.</p> <p>EPCD1: the ERDI-PCD (Enhanced RDI - Path Connectivity Defect) which generates a ""110"" pattern for bits 5, 6 and 7 of the G1 byte.</p> <p>EPPD1: the ERDI-PPD (Enhanced RDI - Path Payload Defect) which generates a ""B010"" pattern for bits 5, 6 and 7 of the G1 byte.</p> <p>LOP: the LOP-P (Loss of Pointer - Path) which generates a non-valid pointer.</p> <p>PDI: the PDI-P (Payload Defect Indication - Path) which generates a Virtual Tributary (VT)-structured Synchronous Transport Signal-Level 1(STS-1) SPE with payload defect.</p> <p>UNEQP1: the UNEQ-P (Unequipped - Path) which generates samples of unequipped Synchronous Transport Signal (STS) signal labels (path and payload are set to ""00 H"").</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:TYPE?</p> <p>Returns: AIS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE?**

<b>Description</b>	<p>This query returns the type of High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to AIS.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer (STS Path) &gt;(Type) Alarms &gt;Defect</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer (AU Path) &gt;(Type) Alarms &gt;Defect</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of High Order Path (HOP) alarm.</p> <p>AIS, Alarm Indication Signal - Path (AIS-P) is selected as High Order Path (HOP) alarm.</p> <p>RDI, Remote Defect Indication - Path (RDI-P) is selected as High Order Path (HOP) alarm.</p> <p>EPSD1, Enhanced Remote Defect Indication - Path Server Defect (ERDI-PSD) is selected as High Order Path (HOP) alarm.</p> <p>EPCD1, Enhanced Remote Defect Indication - Path Connectivity Defect (ERDI-PCD) is selected as High Order Path (HOP) alarm.</p> <p>EPPD1, Enhanced Remote Defect Indication - Path Payload Defect (ERDI-PPD) is selected as High Order Path (HOP) alarm.</p> <p>LOP, Loss Of Pointer - Path (LOP-P) is selected as High Order Path (HOP) alarm.</p> <p>PDI, Payload Defect Indication - Path (PDIP) is selected as High Order Path (HOP) alarm.</p> <p>UNEQP1, Unequipped - Path (UNEQ-P) is selected High Order Path (HOP) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:TYPE?</p> <p>Returns: AIS</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH?

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH**

---

<b>Description</b>	<p>This command enables or disables the High Order Path (HOP) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer (STS Path) &gt; (Type) Alarms &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer (AU Path) &gt; (Type) Alarms &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the HOP (High Order Path) alarm generation.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:TYPE AIS SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH ON SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH? Returns: 1</pre>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH?**

<b>Description</b>	<p>This query returns the status of High Order Path (HOP) alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer (STS Path) &gt;(Type) Alarms &gt;Inject</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer (AU Path) &gt;(Type) Alarms &gt;Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of High Order Path (HOP) alarm generation.</p> <p>1 - alarm generation is enabled.</p> <p>0 - alarm generation is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE**

<b>Description</b>	<p>This command sets the manual error type of High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to B3.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Manual) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Manual) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE <wsp>B3   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B3   REI</p> <p>Selects the type of High Order Path (HOP) error.</p> <p>B3: B3</p> <p>REI: REI-P (Remote Error Indicator - Path)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:MAN:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:MAN:TYPE?</p> <p>Returns: B3</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOUNT?

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE?**

<b>Description</b>	<p>This query returns the manual error type of High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to B3.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt;(Type)Errors&gt;Mode(Manual)&gt;Defect</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt;(Type)Errors&gt;Mode(Manual)&gt;Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of High Order Path (HOP) error.</p> <p>B3, B3 is selected as High Order Path (HOP) error.</p> <p>REI, Remote Error Indication - Path (REIP) is selected as High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:MAN:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:MAN:TYPE?</p> <p>Returns: B3</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMount

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOunt**

<b>Description</b>	<p>This command sets the amount of High Order Path (HOP) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Manual) &gt; Amount</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOunt <wsp> <Amount>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of HOP (High Order Path) error.</p> <p>Choices are 1 through 50. The default setting is 1.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOUnt?

---

<b>Description</b>	<p>This query returns the amount of High Order Path (HOP) error to inject.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt;(Type)Errors&gt;Mode(Manual)&gt;Amount</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt;(Type)Errors&gt;Mode(Manual)&gt;Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOUnt?{<wsp>MAXimum   MINimum}
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE

---



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJect**

<b>Description</b>	<p>This command injects the type of High Order Path (HOP) error.</p> <p>This command is an event and has no associated *RST condition or query form.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt;(Type)Errors &gt;Inject</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt;(Type)Errors &gt;Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE**

<b>Description</b>	<p>This command selects the type of High Order Path (HOP) error for automated injection. At *RST condition, this value is set to B3.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt; (Type)Errors&gt; Mode(Rate/MaxRate) &gt;Defect</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt; (Type)Errors&gt; Mode(Rate/MaxRate) &gt;Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE &lt;wsp&gt;B3   REI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B3   REI</p> <p>Selects the type of High Order Path (HOP) error for automated injection.</p> <p>B3: B3</p> <p>REI: REI-P (Remote Error Indicator - Path)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:TYPE?</p> <p>Returns: B3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE?**

---

<b>Description</b>	<p>This query returns the type of High Order Path (HOP) error for automated injection.</p> <p>At *RST condition, this value is set to B3.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Rate/MaxRate) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Rate/MaxRate) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of High Order Path (HOP) error for the automated injection.</p> <p>B3, B3 is selected as High Order Path (HOP) error.</p> <p>REI, Remote Error Indication - Path (REIP) is selected as High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:TYPE?</p> <p>Returns: B3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated</p>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE**

<b>Description</b>	<p>This command sets the injection rate for the selected High Order Path (HOP) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt; (Type)Errors&gt; Mode(Rate)</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt; (Type)Errors&gt; Mode(Rate)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected High Order Path (HOP) error.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:RATE?</p> <p>Returns: 1.0E-10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated?</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the injection rate for the selected High Order Path (HOP) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Rate)</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Rate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the injected rate is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:RATE?</p> <p>Returns: 1.0E-10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated</p>

## SCPI Command Reference

### Alarms/Errors

---

<b>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated</b>	
<b>Description</b>	<p>This command enables or disables the selected automated High Order Path (HOP) error at the rate specified or continuously.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt; (Type)Errors&gt; Mode(Rate/MaxRate) &gt;Inject</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt; (Type)Errors&gt; Mode(Rate/MaxRate) &gt;Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated High Order Path (HOP) error injection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHS:ERRor:HOP:PATH:AUTomated:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHS:ERRor:HOP:PATH:AUTomated:RATE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated?**

<b>Description</b>	<p>This query returns the status of automated High Order Path (HOP) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Rate/MaxRate) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Rate/MaxRate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of automated High Order Path (HOP) error injection.</p> <p>1 - automated HOP error injection is enabled</p> <p>0 - automated HOP error injection is disabled</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:CONTInuous**

<b>Description</b>	<p>This command enables or disables the continuously rate of automated High Order Path (HOP) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt; (Type)Errors&gt; Max Rate</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt; (Type)Errors&gt; Max Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the rate of automated High Order Path (HOP) error injection as continuously.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:CONT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE?</p>



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:CONTInuous?**

<b>Description</b>	<p>This query returns the status of continuously rate of automated High Order Path (HOP) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Max Rate</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Max Rate</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of continuously rate of automated High Order Path (HOP) error injection.</p> <p>1 - automated HOP continuous mode is enabled</p> <p>0 - automated HOP continuous mode is disabled</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:CONT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated</p>

## SCPI Command Reference

### Alarms/Errors

---

:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE	
<b>Description</b>	<p>This command selects the Burst error type for the High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to B3.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)&gt;Defect</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)&gt;Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE <wsp>B3   REI
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: B3   REI</p> <p>Selects the Burst error type for the High Order Path (HOP) error.</p> <p>B3: B3 (ERROR)</p> <p>REI: Remote Error Indication - Path (REI-P)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:TYPE?</p> <p>Returns: B3</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:MODE?

---

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERROR:HOP:PATH:BURSt:TYPE?**

---

<b>Description</b>	<p>This query returns the Burst error type for the High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to B3.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERROR:HOP:PATH:BURSt:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Burst type for the High Order Path (HOP) error.</p> <p>B3, B3 (ERROR) is selected as HOP error.</p> <p>REI, Remote Error Indication - Path (REI) is selected as HOP error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:TYPE B3</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:TYPE?</p> <p>Returns: B3</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDHSonet:ERROR:HOP:PATH:BURSt:MODE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:MODE**

<b>Description</b>	<p>This command selects the Burst mode for the High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt;(Type)Errors &gt;Mode(Burst Single/Burst Repeat)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:MODE &lt;wsp&gt;SINGLE   REPEAT</p>
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SINGLE   REPEAT</p> <p>Selects the Burst mode for the High Order Path (HOP) error.</p> <p>SINGLE: Single</p> <p>REPEAT: Repeat</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:MODE SINGLE</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:MODE?</p> <p>Returns: SINGLE</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:MODE?**

<b>Description</b>	<p>This query returns the Burst mode for the High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Burst mode for the High Order Path (HOP) error.</p> <p>SINGLE, Single is selected as Burst mode.</p> <p>REPEAT, Repeat is selected as Burst mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:MODE SINGLE</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:MODE?</p> <p>Returns: SINGLE</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:DURation**

<b>Description</b>	<p>This command sets the burst duration for the High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:DURation &lt;wsp&gt; &lt;Duration&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Duration:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the burst duration for the High Order Path (HOP) error.</p> <p>Choices are 1 through 14400000 or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:DURation?**

<b>Description</b>	<p>This query returns the burst duration for the High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:DURation? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current duration is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Duration>
<b>Response(s)</b>	<p><b>Duration:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the burst duration for the High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:PERiod

---

<b>Description</b>	<p>This command sets the burst period for the High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Burst Repeat) &gt; Period</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:PERiod &lt;wsp&gt; &lt;Period&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Period:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the burst period for the High Order Path (HOP) error.</p> <p>Choices are 1 through 14400000 or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:PER 15 SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:PER? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE? SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:MODE?</pre>

---



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:PERiod?**

<b>Description</b>	<p>This query returns the burst period for the High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Burst Repeat) &gt; Period</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:PERiod? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current period is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Period>
<b>Response(s)</b>	<p><b>Period:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the burst period for the High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:PER 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS:PER?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:MODE</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt

---

<b>Description</b>	<p>This command enables or disables the Burst generation for the High Order Path (HOP) error. At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Burst generation for the High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt?**

<b>Description</b>	<p>This query returns the status of Burst generation for the High Order Path (HOP) error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type)Errors &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Burst generation for the High Order Path (HOP) error.</p> <p>1 - automated HOP ERROR burst injection enabled</p> <p>0 - automated HOP ERROR burst injection is disabled</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:PATH:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:BURSt:TYPE

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE**

<b>Description</b>	<p>This command selects the burst alarm type for the High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to AIS.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE &lt;wsp&gt;AIS   UNEQP1   RDI   PDI   LOP   EPSD1   EPCD1   EPPD1</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AIS   UNEQP1   RDI   PDI   LOP   EPSD1   EPCD1   EPPD1</p> <p>Selects the burst alarm type for the High Order Path (HOP) alarm.</p> <p>AIS: Administrative Unit - Alarm Indication Signal (AIS)</p> <p>HPRDi: High Order path - Remote Defect Indication</p> <p>EPSD1: Enhanced RDI - Server Defect (EPSD1)</p> <p>EPCD1: Enhanced RDI - Connectivity Defect (EPCD1)</p> <p>EPPD1: Enhanced RDI - Payload Defect (EPPD1)</p> <p>AULop: Administrative Unit - Loss of Pointer (AULOP)</p> <p>UNEQP1: High Order Path - Unequipped (UNEQ-P)</p> <p>PDI: High Order Path - PDI</p> <p>RDI: High Order Path - RDI</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:TYPE?</p> <p>Returns: AIS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE?**

<b>Description</b>	<p>This query returns the burst alarm type for the High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to AIS.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the burst alarm type for the High Order Path (HOP) alarm.</p> <p>AIS, Administrative Unit - Alarm Indication Signal (AIS) is selected as HOP alarm.</p> <p>RDI, High Order path - Remote Defect Indication (RDI) is selected as HOP alarm.</p> <p>EPSD1, Enhanced RDI - Server Defect (EPSD1) is selected as HOP alarm.</p> <p>EPCD1, Enhanced RDI - Connectivity Defect (EPCD1) is selected as HOP alarm.</p> <p>EPPD1, Enhanced RDI - Payload Defect (EPPD1) is selected as HOP alarm.</p> <p>LOP, Administrative Unit - Loss of Pointer (LOP) is selected as HOP alarm.</p> <p>UNEQP1, High Order Path - Unequipped (UNEQ-P) is selected as HOP alarm.</p> <p>PDI, High Order Path - PDI is selected as HOP alarm.</p> <p>LOM, High Order Path - LOM is selected as HOP alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:TYPE AIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:TYPE?</p> <p>Returns: AIS</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:MODE

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:MODE

---

<b>Description</b>	<p>This command selects the Burst mode for the High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt;(Type)Alarms &gt;Mode(Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt;(Type)Alarms &gt;Mode(Burst Single/Burst Repeat)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:MODE <wsp>SINGLE   REPEAT
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SINGLE   REPEAT</p> <p>Selects the Burst mode for the High Order Path (HOP) alarm.</p> <p>SINGLE: Single</p> <p>REPEAT: Repeat</p>
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:MODE SINGLE SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:MODE? Returns: SINGLE
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE?

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:MODE?**

---

<b>Description</b>	<p>This query returns the Burst mode for the High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type) Alarms &gt; Mode (Burst Single/Burst Repeat)</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type) Alarms &gt; Mode (Burst Single/Burst Repeat)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Burst mode for the High Order Path (HOP) alarm.</p> <p>SINGLE, Single is selected as Burst mode.</p> <p>REPEAT, Repeat is selected as Burst mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:MODE SINGLE</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:MODE?</p> <p>Returns: SINGLE</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:DURation**

<b>Description</b>	<p>This command sets the burst duration for the High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SONET&gt;Layer(STS Path)&gt;(Type)Alarms &gt;Mode(Burst Single/Burst Repeat)&gt;Duration</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt;Results&gt;Alarms/Errors&gt;SDH&gt;Layer(AU Path)&gt;(Type)Alarms &gt;Mode(Burst Single/Burst Repeat)&gt;Duration</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:DURation &lt;wsp&gt; &lt;Duration&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Duration:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the burst duration for the High Order Path (HOP) alarm.</p> <p>Choices are 1 through 14400000 or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:MODE?</p>



**:SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:DURation?**

<b>Description</b>	<p>This query returns the burst duration for the High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Duration</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:DURation? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current duration is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Duration>
<b>Response(s)</b>	<p><b>Duration:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the burst duration for the High Order Path (HOP) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:DUR 15</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:DUR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:MODE</p>

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:PERiod**

<b>Description</b>	<p>This command sets the burst period for the High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Period</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:PERiod &lt;wsp&gt; &lt;Period&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Period:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the burst period for the High Order Path (HOP) alarm.</p> <p>Choices are 1 through 14400000 or 0.00125 to 1800 seconds.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:PER 15</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:PER?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:MODE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:PERiod?**

---

<b>Description</b>	<p>This query returns the burst period for the High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type) Alarms &gt; Mode (Burst Single/Burst Repeat) &gt; Period</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type) Alarms &gt; Mode (Burst Single/Burst Repeat) &gt; Period</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:PERiod? [<wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current period is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Period>
<b>Response(s)</b>	<p><b>Period:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the burst period for the High Order Path (HOP) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:PER 15</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS:PER?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:PATH:BURSt:MODE</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt

---

<b>Description</b>	<p>This command enables or disables the burst generation for the High Order Path (HOP) alarm. At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the burst generation for the High Order Path (HOP) alarm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE?

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt?**

<b>Description</b>	<p>This query returns the status of Burst for the High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SONET &gt; Layer(STS Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Alarms/Errors &gt; SDH &gt; Layer(AU Path) &gt; (Type) Alarms &gt; Mode(Burst Single/Burst Repeat) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of burst generation for the High Order Path (HOP) alarm.</p> <p>1 - automated HOP ALARM burst injection enabled</p> <p>0 - automated HOP ALARM burst injection is disabled</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:PATH:BURS?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:BURSt:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:TYPE**

<b>Description</b>	<p>This command sets the manual type of High Order Path (HOP) TCM error.</p> <p>At *RST condition, this value is set to LIEC.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (AU PATH TCM) &gt; Type (Errors) &gt; Defect</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (STS PATH TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:TYPE &lt;wsp&gt;LIEC   LOEI   LREI</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIEC   LOEI   LREI</p> <p>Selects the type of TCM error for Higher Rates.</p> <p>LIEC: LIEC</p> <p>LOEI: LOEI</p> <p>LREI: LREI</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:TYPE LOEI</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:TYPE?</p> <p>Returns: LOEI</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOUNT?</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:TYPE?**

---

<b>Description</b>	<p>This query returns the manual type of High Order Path (HOP) TCM error.</p> <p>At *RST condition, this value is set to LIEC.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of High Order Path (HOP) TCM error.</p> <p>LIEC, returns the LIEC as the type of HOP TCM error.</p> <p>LOEI, returns the LOEI as the type of HOP TCM error.</p> <p>LREI, returns the LREI as the type HOP TCM error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:TYPE LOEI</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:TYPE?</p> <p>Returns: LOEI</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOUNT

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:AMOUNT**

<b>Description</b>	<p>This command sets the amount of HOP-TCM error to be injected.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:AMOUNT &lt;wsp&gt; &lt;Amount&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Amount:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the amount of HOP TCM error. Choices are 1 to 50.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE?</p>



---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:AMOUnt?**

<b>Description</b>	<p>This query returns the amount of HOP-TCM error to be injected.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Mode (Manual) &gt; Amount</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:MANual:AMOUnt? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current value of HOP TCM error to inject is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Amount>
<b>Response(s)</b>	<p><b>Amount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the amount of TCM error for Higher rates.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:AMO?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE

---

## SCPI Command Reference

### *Alarms/Errors*

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:INJect**

---

<b>Description</b>	<p>This command injects the type of HOP-TCM error.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Mode (Manual) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:TYPE**

---

<b>Description</b>	<p>This command selects the type of HOP-TCM error for automated injection.</p> <p>At *RST condition, this value is set to LIEC.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:TYPE &lt;wsp&gt;LIEC   LOEI   LREI</pre>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIEC   LOEI   LREI</p> <p>Selects the type of TCM error of Higher Rates for automated injection.</p> <p>LIEC: LIEC</p> <p>LOEI: LOEI</p> <p>LREI: LREI</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:TYPE LOEI SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:TYPE? Returns: LOEI</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE? SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:TYPE?**

<b>Description</b>	<p>This query returns the type of HOP-TCM error for automated injection.</p> <p>At *RST condition, this value is set to LIEC.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:TYPE?
<b>Response Syntax</b>	<Error>
<b>Response(s)</b>	<p><b>Error:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of TCM error for the automated injection.</p> <p>LIEC, LIEC is selected.</p> <p>LOEI, LOEI is selected.</p> <p>LREI, LREI is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:TYPE LOEI</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:TYPE?</p> <p>Returns: LOEI</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:RATE**

---

<b>Description</b>	<p>This command sets the injection rate for the selected HOP-TCM error error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:RATE &lt;wsp&gt; &lt;Rate&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Rate:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the injection rate for the selected TCM error for automated injection.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:RATE 1.0E-10 SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:RATE? Returns: 1.0E-10</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE? SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:RATE?**

<b>Description</b>	<p>This query returns the injection rate for the selected High Order Path (HOP) error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Mode (Rate) &gt; Rate</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:RATE? [ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current injection rate for the selected TCM error is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the injection rate for the selected TCM error for Higher Rates.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:RATE?</p> <p>Returns: 1.0E-10</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated**

---

<b>Description</b>	<p>This command enables or disables the selected automated High Order Path (HOP) TCM error.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Mode (Rate/Max Rate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the automated TCM error injection for Higher Rates</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:RATE 1.0E-10 SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT ON SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE? SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated?

---

<b>Description</b>	<p>This query returns the status of automated HOP-TCM error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Mode (Rate/Max Rate) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of TCM automated error injection for Higher Rates.</p> <p>1, TCM automated error injection is enabled.</p> <p>0, TCM automated error injection is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:RATE 1.0E-10</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:RATE</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:CONTInuous**

---

<b>Description</b>	<p>This command enables or disables the continuously rate of automated High Order Path (HOP) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:CONTInuous &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the rate of TCM error injection as continuous for Higher Rates.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:CONT ON SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:CONT? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE?</pre>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### **:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:CONTInuous?**

---

<b>Description</b>	<p>This query returns the status of the continuous rate of automated High Order Path (HOP) error injection.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Errors) &gt; Mode (Max Rate)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:AUTomated:CONTInuous?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the continuous rate of automated TCM error injection for Higher Rates.</p> <p>1, TCM error injection continuous mode is enabled.</p> <p>0, TCM error injection continuous mode is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:CONT ON</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AUT:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AUTomated</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TYPE**

<b>Description</b>	<p>This command selects the type of High Order Path (HOP) TCM alarm.</p> <p>At *RST condition, this value is set to LLTC.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TYPE <wsp>LIAIS   LLTC   LODI   LRDI   LUNEQ
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIAIS   LLTC   LODI   LRDI   LUNEQ</p> <p>Selects the type of TCM alarm for Higher Rates.</p> <p>LIAIS: LIAIS</p> <p>LLTC: LLTC</p> <p>LODI: LODI</p> <p>LRDI: LRDI</p> <p>LUNEQ: LUNEQ</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM:TYPE LIAIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM:TYPE?</p> <p>Returns: LIAIS</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TYPE?**

<b>Description</b>	<p>This query returns the type of High Order Path (HOP) alarm.</p> <p>At *RST condition, this value is set to LLTC.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Alarms) &gt; Defect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:TYPE?
<b>Response Syntax</b>	<Alarm>
<b>Response(s)</b>	<p><b>Alarm:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of High Order Path (HOP) TCM alarm.</p> <p>LIAIS, LIAIS is selected as High Order Path (HOP) TCM alarm</p> <p>LLTC, LLTC is selected as High Order Path (HOP) TCM alarm</p> <p>LODI, LODI is selected as High Order Path (HOP) TCM alarm</p> <p>LRDI, LRDI is selected as High Order Path (HOP) TCM alarm</p> <p>LUNEQ, LUNEQ is selected as High Order Path (HOP) TCM alarm</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM:TYPE LIAIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM:TYPE?</p> <p>Returns: LIAIS</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH?

**:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARM:HOP:TCM**

<b>Description</b>	<p>This command enables or disables the status of High Order Path (HOP) TCM alarm generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARM:HOP:TCM <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the TCM alarm generation for Higher Rates.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM:TYPE LIAIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDHSONet:ALARM:HOP:PATH:TYPE

---

## SCPI Command Reference

### Alarms/Errors

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM?**

<b>Description</b>	<p>This query returns the status of High Order Path (HOP) error generation.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT &gt; Results &gt; Alarms/Errors &gt; Global Injection &gt; Layer (HOP-TCM) &gt; Type (Alarms) &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the High Order Path (HOP) TCM alarm generation.</p> <p>0, High Order Path (HOP) TCM alarm is disabled.</p> <p>1, High Order Path (HOP) TCM alarm is enabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM:TYPE LIAIS</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM ON</p> <p>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE?

**:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:HISTory?**

<b>Description</b>	<p>This query returns the history status of the High Order Path (HOP) TCM error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; HOP-TCM</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:SDHSONet:ERRor:HOP:TCM:HISTory? &lt;wsp&gt;LIEC   LOEI   LREI   LVIOL</code>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIEC   LOEI   LREI   LVIOL</p> <p>Selects the type of TCM error for Higher Rates.</p> <p>LIEC: LIEC</p> <p>LOEI: LOEI</p> <p>LREI: LREI</p> <p>LVIOL: LVIOL</p>
<b>Response Syntax</b>	<code>&lt;History&gt;</code>

---

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:HISTory?

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of High Order Path (HOP) error. PRESENT, indicates that at least one error has occurred. ABSENT, indicates that no error occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:TYPE LOEI SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AMO 15 SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:INJ FETC:DATA:TEL:SDHS:ERR:HOP:TCM:HIST? LIEC
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOUNT SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJECT

---



---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds within which the High Order Path (HOP) TCM error occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; HOP-TCM</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:SEConds? <wsp>LIEC   LOEI   LREI   LVIOL
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIEC   LOEI   LREI   LVIOL</p> <p>Selects the type of TCM error for Higher Rates.</p> <p>LIEC: LIEC</p> <p>LOEI: LOEI</p> <p>LREI: LREI</p> <p>LVIOL: LVIOL</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds within which the selected test error occurred.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:TYPE LOEI</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:HOP:TCM:SEC? LIEC</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMount</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJect</p>

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:CURRent?

---

<b>Description</b>	<p>This query returns the current status of High Order Path (HOP) TCM error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; HOP-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:CURRent? &lt;wsp&gt;LIEC   LOEI   LREI   LVIOL</p>
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIEC   LOEI   LREI   LVIOL</p> <p>Selects the type of TCM error for Higher Rates.</p> <p>LIEC: LIEC</p> <p>LOEI: LOEI</p> <p>LREI: LREI</p> <p>LVIOL: LVIOL</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:CURRent?****Response(s)****Current:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current status of High Order Path (HOP) error.

PRESENT, indicates that at least one error has occurred in the last second.

ABSENT, indicates that there is no error.

INACTIVE, indicates that the test is not running.

**Example(s)**

SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:TYPE LOEI

SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AMO 15

SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:INJ

FETC:DATA:TEL:SDHS:ERR:HOP:TCM:CURR? LIEC

**See Also**

SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE

SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOut

SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJect

---

## SCPI Command Reference

### Alarms/Errors

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:COUNT?

<b>Description</b>	<p>This query returns the count of High Order Path (HOP) TCM error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; HOP-TCM</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:COUNT? <wsp>LIEC   LOEI   LREI   LVIOL
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIEC   LOEI   LREI   LVIOL</p> <p>Selects the type of TCM error for Higher Rates.</p> <p>LIEC: LIEC</p> <p>LOEI: LOEI</p> <p>LREI: LREI</p> <p>LVIOL: LVIOL</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of High Order Path (HOP) error.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:MAN:TYPE LOEI</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:AMO 15</p> <p>SOUR:DATA:TEL:SDHS:ERR:HOP:TCM:INJ</p> <p>FETC:DATA:TEL:SDHS:ERR:HOP:TCM:COUN? LIEC</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:AMOUNT</p> <p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:PATH:INJect</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:RATE?**

<b>Description</b>	<p>This query returns the current rate of High Order Path (HOP) TCM error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; HOP-TCM</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ERRor:HOP:TCM:RATE? <wsp>LIEC   LOEI   LREI   LVIOL
<b>Parameter(s)</b>	<p><b>Error:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIEC   LOEI   LREI   LVIOL</p> <p>Selects the type of TCM error for Higher Rates.</p> <p>LIEC: LIEC</p> <p>LOEI: LOEI</p> <p>LREI: LREI</p> <p>LVIOL: LVIOL</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of High Order Path (HOP) error.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ERR:HOP:TCM:RATE? LIEC
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHS:ERRor:HOP:PATH:MANual:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:SDHS:ERRor:HOP:PATH:AMOUnt</p> <p>SOURce[1..n]:DATA:TELEcom:SDHS:ERRor:HOP:PATH:INJect</p>

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELecom:SDHSonet:ALARm:HOP:TCM:HISTory?

---

<b>Description</b>	<p>This query returns the history status of the High Order Path (HOP) TCM alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; HOP-TCM</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:SDHSonet:ALARm:HOP:TCM:HISTory? &lt;wsp&gt;LIAIS   LLTC   LODI   LRDI   LUNEQ   LTIM</p>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIAIS   LLTC   LODI   LRDI   LUNEQ   LTIM</p> <p>Selects the type of TCM alarm for Higher Rates.</p> <p>LIAIS: LIAIS</p> <p>LLTC: LLTC</p> <p>LODI: LODI</p> <p>LRDI: LRDI</p> <p>LUNEQ: LUNEQ</p> <p>LTIM: LTIM</p>
<b>Response Syntax</b>	<p>&lt;History&gt;</p>

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:HISTory?**

<b>Response(s)</b>	<b>History:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of High Order Path (HOP) alarm. PRESENT, indicates that at least one alarm has occurred. ABSENT, indicates that no alarm occurred. INACTIVE, indicates that the test did not run yet.
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM:TYPE LIAIS SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM ON FETC:DATA:TEL:SDHS:ALAR:HOP:TCM:HIST? LIAIS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH

---

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which the High Order Path (HOP) TCM alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; HOP-TCM</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:SEConds? &lt;wsp&gt;LIAIS   LLTC   LODI   LRDI   LUNEQ   LTIM</code>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIAIS   LLTC   LODI   LRDI   LUNEQ   LTIM</p> <p>Selects the type of TCM alarm for Higher Rates.</p> <p>LIAIS: LIAIS</p> <p>LLTC: LLTC</p> <p>LODI: LODI</p> <p>LRDI: LRDI</p> <p>LUNEQ: LUNEQ</p> <p>LTIM: LTIM</p>
<b>Response Syntax</b>	<code>&lt;Seconds&gt;</code>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds within which the selected test alarm occurred.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM:TYPE LIAIS SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM ON FETC:DATA:TEL:SDHS:ALAR:HOP:TCM:SEC? LIAIS</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH</pre>

---



**:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM:CURRent?**

<b>Description</b>	<p>This query returns the current status of the High Order Path (HOP) TCM alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; SONET/SDH BERT&gt; Results &gt; Alarms/Errors &gt; HOP-TCM</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:SDHSONet:ALARm:HOP:TCM:CURRent? &lt;wsp&gt;LIAIS   LLTC   LODI   LRDI   LUNEQ   LTIM</code>
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LIAIS   LLTC   LODI   LRDI   LUNEQ   LTIM</p> <p>Selects the type of TCM alarm for Higher Rates.</p> <p>LIAIS: LIAIS</p> <p>LLTC: LLTC</p> <p>LODI: LODI</p> <p>LRDI: LRDI</p> <p>LUNEQ: LUNEQ</p> <p>LTIM: LTIM</p>
<b>Response Syntax</b>	<code>&lt;Current&gt;</code>

---

## SCPI Command Reference

### Alarms/Errors

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:TCM:CURRent?**

<b>Response(s)</b>	<b>Current:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of High Order Path (HOP) alarm. PRESENT, indicates that at least one alarm has occurred in the last second. ABSENT, indicates that there is no alarm. INACTIVE, indicates that the test is not running.
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM:TYPE LIAIS SOUR:DATA:TEL:SDHS:ALAR:HOP:TCM ON FETC:DATA:TEL:SDHS:ALAR:HOP:TCM:CURR? LIAIS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH:TYPE SOURce[1..n]:DATA:TELEcom:SDHSonet:ALARm:HOP:PATH

---

**:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:SEConds?**

<b>Description</b>	<p>This query returns the no of seconds within which the global CAUI Lane alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE/Serial(OTU3/OTU3E1/OTU3E2)/STM-256/OC-768 &gt;Results &gt; Alarm/Errors &gt; Interface</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:SEConds? <wsp>INTerlinkstatus   SERialfreq
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: INTerlinkstatus   SERialfreq</p> <p>Selects the no of seconds within which global CAUI Lane alarm occurred.</p> <p>INTerlinkstatus: INTerlinkstatus</p> <p>SERialfreq: SERialfreq</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the no of seconds within which global CAUI alarm occurred.</p>
<b>Example(s)</b>	FETC:DATA:TEL:CAUI:ALAR:GLOB:SEC? INTerlinkstatus
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:CURRent?

## SCPI Command Reference

### Alarms/Errors

---

---

#### :FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:SEConds?

---

<b>Description</b>	<p>This query returns the no of seconds within which an optical port alarm occurred.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; 100GE/40GE/Serial(OTU3/OTU3E1/OTU3E2)/OC-768/STM-256&gt;Results &gt; Alarm/Errors &gt; Interface</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:GLOBal:SEConds? <wsp>LOS
<b>Parameter(s)</b>	<p><b>Alarm:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOS</p> <p>Selects the no of seconds within which optical port global alarm occurred.</p> <p>LOS</p>
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the no of seconds within which optical port alarm occurred</p>
<b>Example(s)</b>	FETC:DATA:TEL:OPT:ALAR:PORT:GLOB:SEC? LOS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:HISTory?

---

---

**:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:TX:STATus?**

---

<b>Description</b>	<p>This query returns the current status of global CAUI Lanes alarm.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Setup &gt; TX/RX</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT &gt; Test Configurator &gt; Setup &gt; TX/RX</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:CAUI:ALARm:GLOBal:TX:STATus?
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Current status of CAUI alarm.</p> <p>PRESENT, indicates that at least one alarm has occurred.</p> <p>ABSENT, indicates that no alarm occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
<b>Example(s)</b>	FETC:DATA:TEL:CAUI:ALAR:GLOBal:TX:STATus?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OPTical:MODule:STATus?

---

## Traces - SONET/SDH

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:COPI**

---

<b>Description</b>	<p>This command sets the copy Rx to instrument for Path Overhead.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test&gt; OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Results &gt; Traces &gt;SONET&gt;Traces &gt;TIM-P&gt;Copy RX</p> <p>Navigation Path: Test&gt; OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Results &gt; Traces &gt;SDH&gt;Traces &gt;HP-TIM&gt;Copy RX</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:COPI</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:PATH:OVER:TIM:COPI</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:COPI</p>

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:POVerhead:J[1..n]:TIM:PATtern:RECeived?**

<b>Description</b>	<p>This query returns the received J1 value for Path Overhead.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SONET &gt; Traces &gt; STS Path (J1) &gt; Received</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SDH &gt; Traces &gt; AU Path (J1) &gt; Received</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:POVerhead:J[1..n]:TIM:PATtern:RECeived?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the received J1 value for Path Overhead.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:POV:J1:TIM:PATT:REC?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHSonet:SOVerhead:J[1..n]:TIM:PATtern:RECeived

---

## SCPI Command Reference

### Traces - SONET/SDH

---

---

#### :SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:COPI

---

<b>Description</b>	<p>This command sets the copy Rx to instrument for Section Overhead.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test&gt; OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Results &gt; Traces &gt;SONET&gt;Traces &gt;TIM-S&gt;Copy RX</p> <p>Navigation Path: Test&gt; OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Results &gt; Traces &gt;SDH&gt;Traces &gt;RS-TIM&gt;Copy RX</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:SECTion:OVERhead:TIM:COPI
<b>Example(s)</b>	SENS:DATA:TEL:SDHS:SECT:OVER:TIM:COPI
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:PATH:OVERhead:TIM:COPI

---



**:FETCh[1..n]:DATA:TELEcom:SDHSonet:SOVerhead:J[1..n]:TIM:PATtern:RECeived?**

<b>Description</b>	<p>This query returns the received J0 value for Section Overhead.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SONET &gt; Traces &gt; Section (J0) &gt; Received</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Test Configurator &gt; Results &gt; Traces &gt; SDH &gt; Traces &gt; RS (J0) &gt; Received</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:SOVerhead:J[1..n]:TIM:PATtern:RECeived?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the received J0 value Section Overhead.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:SOV:J100:TIM:PATT:REC?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHSonet:POVerhead:J[1..n]:TIM:PATtern:RECeived

---

## SCPI Command Reference

### Traces - SONET/SDH

---

---

#### :FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:RECeived?

---

<b>Description</b>	<p>This query returns the received value for Path Overhead.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;OC&gt;Traces&gt; &gt;TCM Access Point Identifier&gt;STS Path&gt;Received</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Test Configurator &gt; Results&gt;Traces &gt;STM&gt;Traces&gt; &gt;TCM Access Point Identifier&gt;AU Path&gt;Received</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:TCAPident:N[1..n]:RECeived?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the received 16 or 64 byte format TCM Access Point Identifier message for Path Overhead.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDHS:HOP:TCAP:N1:RECeived?</p> <p>Returns: "tcmmessage"</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHSonet:POVerhead:J[1..n]:TIM:PATtern:RECeived

---

---

**:SENSe[1..n]:DATA:TELecom:SDHSonet:HOP:TCAPident:COPY**

---

<b>Description</b>	<p>This event sets the copy Rx to instrument for HOP TCM Path Overhead.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test&gt; SONET/SDH BERT&gt; Results &gt; Traces &gt;OC&gt;TIM-P&gt;Copy RX</p> <p>Navigation Path: Test&gt; SONET/SDH BERT&gt; Results &gt; Traces &gt;STM&gt;HPTC-TIM&gt;Copy RX</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:SDHSonet:HOP:TCAPident:COPY
<b>Example(s)</b>	SENS:DATA:TEL:SDHS:HOP:TCAP:COPY
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:SDHSonet:LINE:OVERhead:TIM:COPY

---

## Traces - OTN

---

:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?

---

<b>Description</b>	<p>This query returns the received message for the instrument.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU &gt; TCM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for ODU TCM TTI.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ODU1:TCM1:TTI:SAPI:B?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:B?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:B?**

---

<b>Description</b>	<p>This query returns the received message for ODU TCM TTI.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU &gt; TCM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for ODU TCM TTI.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ODU1:TCM1:TTI:DAPI:B?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?

---

## SCPI Command Reference

### Traces - OTN

---

---

#### :FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:OPSPec:B?

---

<b>Description</b>	<p>This query returns the received message for ODU TCM TTI.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU &gt; TCM TTI Traces &gt; Operator Specific</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for ODU TCM TTI.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ODU1:TCM1:TTI:OPSP:B?
<b>Note(s)</b>	FETCh:DATA:TELecom:OTN:ODU:TCM:TTI:OPSPec:B?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:B?**

---

<b>Description</b>	<p>This query returns the received message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU3e(1/2) &gt; TCM TTI Traces &gt; SAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU(1/2)e &gt; TCM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:SAPI:B?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:B?

---

## SCPI Command Reference

### Traces - OTN

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:B?**

<b>Description</b>	<p>This query returns the received message for the instrument.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU3e(1/2) &gt; TCM TTI Traces &gt; DAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU(1/2)e &gt; TCM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:DAPI:B?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:B?

---



---

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:OPSPec:B?**

---

<b>Description</b>	<p>This query returns the received message for the instrument.</p> <p>At *RST condition, this value of OSPECIFIC set to ""EXFO OTU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU3e(1/2) &gt; TCM TTI Traces &gt; Operator Specific</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU(1/2)e &gt; TCM TTI Traces &gt; Operator Specific</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for the instrument.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:OPSP:B?</p> <p>Returns: "EXFO OTU OPERATOR SPECIFIC"</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:B?

---

## SCPI Command Reference

### Traces - OTN

---

---

#### :FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:SAPI:B?

---

<b>Description</b>	<p>This query returns the received message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>In Multi-Channel OTN, when accessing an ODU LO, the channel must be set using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN.</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for ODU TTI.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU1:TTI:SAPI:B?</p> <p>For Multi-Channel OTN:</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:CHAN 2</p> <p>FETC:DATA:TEL:OTN:ODU100:TTI:SAPI:B?</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?</p> <p>SENS[1..n]:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN?</p>

---

---

**:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:DAPI:B?**

<b>Description</b>	<p>This query returns the received message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>In Multi-Channel OTN, when accessing an ODU LO, the channel must be set using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN.</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for ODU TTI.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU1:TTI:DAPI:B?</p> <p>For Multi-Channel OTN:</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:CHAN 2</p> <p>FETC:DATA:TEL:OTN:ODU100:TTI:DAPI:B?</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?</p> <p>SENS[1..n]:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN?</p>

---

## SCPI Command Reference

### Traces - OTN

---

**:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:OPSPec:B?**

<b>Description</b>	<p>This query returns the received message for ODU TTI Trace.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>In Multi-Channel OTN, when accessing an ODU LO, the channel must be set using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN.</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for ODU TTI.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU1:TTI:OPSP:B?</p> <p>For Multi-Channel OTN:</p> <p>SENS:DATA:TEL:OTN:ODU100:TTI:CHAN 2</p> <p>FETC:DATA:TEL:OTN:ODU100:TTI:OPSP:B?</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?</p> <p>SENS[1..n]:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN?</p>

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:B?**

<b>Description</b>	<p>This query returns the received SAPI message for Over Clocked ODU TTI.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces &gt; SAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for over clocked ODU TTI.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:OTN:ODU3:E1:TTI:SAPI:B?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?

---

## SCPI Command Reference

### Traces - OTN

---

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:B?**

<b>Description</b>	<p>This query returns the received DAPI message for Over Clocked ODU TTI.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces &gt; DAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for over clocked ODU TTI.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:OTN:ODU3:E1:TTI:DAPI:B?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:OPSPec:B?**

<b>Description</b>	<p>This query returns the received OPSPec message for Over Clocked ODU TTI.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU3e(1/2) &gt; PM TTI Traces &gt; Operator Specific</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU(1/2)e &gt; PM TTI Traces &gt; Operator Specific</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TTI:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for over clocked ODU TTI.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:OTN:ODU3:E1:TTI:OPSPec:B?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:B?

---

## SCPI Command Reference

### Traces - OTN

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:B?**

---

<b>Description</b>	<p>This query returns the received message for overclocked rates.</p> <p>At *RST condition, this value is set to ""EXFO OTU SAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces &gt; SAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU(1/2)e &gt; SM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:OTU3:E1:TTI:SAPI:B?</p> <p>Returns: "EXFO OTU SAPI"</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:B?

---



**:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:B?**

<b>Description</b>	<p>This query returns the received message for overclocked rates.</p> <p>At *RST condition, this value is set to ""EXFO OTU DAPI"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces &gt; DAPI</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU(1/2)e &gt; SM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:OTU3:E1:TTI:DAPI:B?</p> <p>Returns: "EXFO OTU DAPI"</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:B?

---

## SCPI Command Reference

### Traces - OTN

---

**:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:OPSPec:B?**

<b>Description</b>	<p>This query returns the received message for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is set to ""EXFO OTU OPERATOR SPECIFIC"".</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU3e(1/2) &gt; SM TTI Traces &gt; Operator Specific</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU(1/2)e &gt; SM TTI Traces &gt; Operator Specific</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:OTU3:E1:TTI:OPSP:B?</p> <p>Returns: "EXFO OTU OPERATOR SPECIFIC"</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:B?

---

---

**:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:B?**

---

<b>Description</b>	<p>This query returns the received message for the instrument.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for OTU.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:OTU1:TTI:SAPI:B?
<b>Note(s)</b>	FETCh:DATA:TELEcom:OTN:OTU:TTI:SAPI:B16?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:OPSPec:B?

---

## SCPI Command Reference

### Traces - OTN

---

---

#### :FETCh[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:DAPI:B?

---

<b>Description</b>	<p>This query returns the received message for the instrument.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:DAPI:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for OTU.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:OTU1:TTI:DAPI:B?
<b>Note(s)</b>	FETCh:DATA:TELecom:OTN:OTU:TTI:DAPI:B16?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:OPSPec:B?

---

**:FETCh[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:OPSPec:B?**

<b>Description</b>	<p>This query returns the received message for the instrument.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces &gt; Operator Specific</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:OPSPec:B?
<b>Response Syntax</b>	<Message>
<b>Response(s)</b>	<p><b>Message:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the selected message for OTU.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:OTU1:TTI:OPSP:B?
<b>Note(s)</b>	FETCh:DATA:TELecom:OTN:OTU:TTI:OPSPec:B32?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:OPSPec:B?

---

## SCPI Command Reference

### Traces - OTN

---

**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:SAPI:COPY**

<b>Description</b>	<p>This command sets the copy Rx to instrument.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test&gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:SAPI:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OTU1:TTI:SAPI:COPY
<b>Note(s)</b>	DATA
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:DAPI:COPY

---

**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:DAPI:COPI**

<b>Description</b>	<p>This command sets the copy Rx to instrument.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test&gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:DAPI:COPI
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OTU1:TTI:DAPI:COPI
<b>Note(s)</b>	SENSe:DATA:TELecom:OTN:OTU:TTI:SAPI:EXPEcted
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:SAPI:COPI

---

## SCPI Command Reference

### Traces - OTN

---

**:SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:SAPI:COPI**

<b>Description</b>	<p>This command copies the received SAPI ODU TTI Trace to the expected ODU TTI Trace. This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>NOTE: In Multi-Channel OTN, when copying an ODU LO Trace, the channel must be set using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN. The Trace can also be copied on all channels by using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:ACHA.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:SAPI:COPI
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU1:TTI:SAPI:COPI
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:DAPI:COPI SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:CHAN SENSe[1..n]:DATA:TELeom:OTN:ODU[1..n]:TTI:ACHA

---



**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:DAPI:COPY**

<b>Description</b>	<p>This command copies the received DAPI ODU TTI Trace to the expected ODU TTI Trace.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; TTI Traces &gt; ODU &gt; PM TTI Traces</p> <p>NOTE: In Multi-Channel OTN, when copying an ODU LO Trace, the channel must be set using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:CHAN. The Trace can also be copied on all channels by using SENS:DATA:TEL:OTN:ODU[1..n]:TTI:ACHA.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:DAPI:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU1:TTI:DAPI:COPY
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:SAPI:COPY SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:CHAN SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:ACHA

---

## SCPI Command Reference

### Traces - OTN

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:COPY**

<b>Description</b>	<p>This command sets the copy Rx to instrument.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test&gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:SAPI:COPY
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:COPY

---

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:COPY**

<b>Description</b>	<p>This command sets the copy Rx to instrument.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:DAPI:COPY
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:COPY

---

## SCPI Command Reference

### Traces - OTN

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:COPY**

<b>Description</b>	This command sets the copy Rx to instrument for over clocked rates [ example OTU3e(1/2)]. This command is an event and is not associated with an *RST condition or a query form. Navigation Path: Test> OTN BERT > Results > TTI Traces > OTU > SM TTI Traces > SAPI
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E[1..n]:TTI:SAPI:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OTU3:E1:TTI:SAPI:COPY
<b>Note(s)</b>	DATA
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:COPY

---

**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:COPY**

<b>Description</b>	<p>This command sets the copy Rx to instrument for over clocked rates [ example OTU3e(1/2)]. This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test&gt; OTN BERT &gt; Results &gt; TTI Traces &gt; OTU &gt; SM TTI Traces &gt; DAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:E[1..n]:TTI:DAPI:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OTU3:E1:TTI:DAPI:COPY
<b>Note(s)</b>	SENSe:DATA:TELecom:OTN:OTU:TTI:SAPI:EXPeCted
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:TTI:SAPI:COPY

---

## SCPI Command Reference

### Traces - OTN

---

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:COPY**

<b>Description</b>	This command sets the copy Rx to instrument for over clocked rates [ example OTU3e(1/2)]. This command is an event and is not associated with an *RST condition or a query form. Navigation Path: Test> OTN BERT > Results > TTI Traces > ODU > PM TTI Traces > SAPI
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E[1..n]:TTI:SAPI:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU3:E1:TTI:SAPI:COPY
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:DAPI:COPY

---

---

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:COPY**

---

<b>Description</b>	This command sets the copy Rx to instrument for over clocked rates [ example OTU3e(1/2)]. This command is an event and is not associated with an *RST condition or a query form. Navigation Path: Test> OTN BERT > Results > TTI Traces > ODU > PM TTI Traces > DAPI
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E[1..n]:TTI:DAPI:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU3:E1:TTI:DAPI:COPY
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:SAPI:COPY

---

## SCPI Command Reference

### Traces - OTN

---

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:COPY**

---

<b>Description</b>	<p>This command sets the copy Rx to instrument for over clocked rates [ example OTU3e(1/2)].</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test&gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:SAPI:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:SAPI:COPY
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:COPY

---



---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:COPY**

---

<b>Description</b>	<p>This command sets the copy Rx to instrument for over clocked rates [ example OTU3e(1/2)].</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; TTI Traces &gt; ODU &gt; ODU-TCM &gt; TCM TTI Traces &gt; SAPI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:TCM[1..n]:TTI:DAPI:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU3:E1:TCM1:TTI:DAPI:COPY
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:COPY

---

## Logger and Alarms/Errors Logger

---

:FETCh[1..n]:DATA:TELEcom:LOGGer:EVENTs?

---

<b>Description</b>	This query returns the total number of test events recorded. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Logger
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:LOGGer:EVENTs?
<b>Response Syntax</b>	<Event>
<b>Response(s)</b>	<b>Event:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the total number of test events recorded.
<b>Example(s)</b>	FETC:DATA:TEL:LOGG:EVEN?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:LOGGer:LIST?

---

---

**:FETCh[1..n]:DATA:TELecom:LOGGer:LIST?**

---

<b>Description</b>	<p>This query returns the list of test events.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Logger</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELecom:LOGGer:LIST?[ &lt;wsp&gt; &lt;Eventno&gt;]</code>
<b>Parameter(s)</b>	<p><b>Eventno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>If event number is not provided, then the entire event list is returned and if it is provided the event of that particular number from the list is returned.</p>
<b>Response Syntax</b>	<code>&lt;List&gt;</code>
<b>Response(s)</b>	<p><b>List:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the list of test events.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:LOGG:LIST?</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELecom:LOGGer:EVENTs?</code>

---

## Performance Monitoring

:FETCh[1..n]:DATA:TELEcom:PATtern:PM:STATistics?

<b>Description</b>	<p>This query returns the performance monitoring statistics of Pattern.</p> <p>At *RST condition, this value</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Result &gt; Performance Monitoring &gt; Bert</p> <p>Navigation Path: Test Setup &gt; OTN -SONET /SDH BERT &gt; Result &gt; Performance Monitoring &gt; Bert</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:PATtern:PM:STATistics? &lt;wsp&gt;G821   M2100OOSM, EFS   EC   ES   SES   UAS   ESR   SESR   DM</p>
<b>Parameter(s)</b>	<p><b>Standard:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: G821   M2100OOSM</p> <p>Selects the performance monitoring standard number.</p> <p>G821: G.821</p> <p>M2100OOSM: M.2100 OOSM</p> <p><b>Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: EFS   EC   ES   SES   UAS   ESR   SESR   DM</p> <p>Selects the performance monitoring statistics.</p> <p>EFS: Error Free Seconds (EFS).</p> <p>EC: Error Count (EC).</p> <p>ES: Errored Seconds (ES).</p> <p>SES: Severely Errored Seconds (SES).</p> <p>UAS: Unavailable Second (UAS).</p> <p>ESR: Errored Second Ratio (ESR).</p> <p>SESR: Severely Errored Second Ratio (SESR).</p> <p>DM: Degraded Minutes (DM).</p>
<b>Response Syntax</b>	<p>&lt;Statistics&gt;</p>

---

**:FETCh[1..n]:DATA:TELEcom:PATtern:PM:STATistics?**

---

<b>Response(s)</b>	<b>Statistics:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns performance monitoring statistics of Pattern.
<b>Example(s)</b>	FETC:DATA:TEL:PATT:PM:STAT? G821, EFS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHSONet:SECTion:PM:STATistics?

---

## SCPI Command Reference

### *Performance Monitoring*

---

---

**:FETCh[1..n]:DATA:TELEcom:SONet:SECTion:PM:STATistics?**

---

<b>Description</b>	<p>This query returns the performance monitoring statistics of Section.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN -SONET /SDH BERT &gt; Result &gt; Performance Monitoring &gt; Section &gt; OTN SONET/SDH BERT &gt; Result &gt; Performance Monitoring &gt; RS</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:SONet:SECTion:PM:STATistics? &lt;wsp&gt;G829ISM, EFS   EB   ES   SES   BBE   UAS   ESR   SESR   BBER, NEND</p>

---

**:FETCh[1..n]:DATA:TELEcom:SONet:SECTION:PM:STATistics?**

<b>Parameter(s)</b>	<p><b>Standard:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: G829ISM</p> <p>Selects the performance monitoring standard number.</p> <p>G829ISM: G.829 ISM</p> <p><b>Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: EFS   EB   ES   SES   BBE   UAS   ESR   SESR   BBER</p> <p>Selects the performance monitoring statistics.</p> <p>EFS: Error Free Seconds (EFS).</p> <p>EB: Errored Block (EB).</p> <p>ES: Errored Seconds (ES).</p> <p>SES: Severely Errored Seconds (SES).</p> <p>BBE: Background Block Error (SES).</p> <p>UAS: Unavailable Second (UAS).</p> <p>ESR: Errored Second Ratio (ESR).</p> <p>SESR: Severely Errored Second Ratio (SESR).</p> <p>BBER: Background Block Error Ratio (BBER).</p> <p><b>End:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NEND</p> <p>Selects the Near-End.</p> <p>NEND: the standard for Near-End.</p>
---------------------	--

**Response Syntax** <Statistics>

## SCPI Command Reference

### *Performance Monitoring*

---

---

#### **:FETCh[1..n]:DATA:TELEcom:SONet:SECTion:PM:STATistics?**

<b>Response(s)</b>	<b>Statistics:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the performance monitoring statistics of Section.
<b>Example(s)</b>	FETC:DATA:TEL:SON:SECT:PM:STAT? G829ISM,EFS,NEND
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHSONet:HOP:PM:STATistics?

---



---

**:FETCh[1..n]:DATA:TELEcom:SONet:LINE:PM:STATistics?**

---

**Description**

This query returns the performance monitoring statistics of Line.

At \*RST condition, this value is device dependent.

Navigation Path: Test Setup > OTN -SONET /SDH BERT > Result > Performance Monitoring  
> Line > OTN SONET/SDH BERT > Result > Performance Monitoring > MS

**Syntax**

:FETCh[1..n]:DATA:TELEcom:SONet:LINE:PM:STATistics? <wsp>G829ISM |  
M2101ISM, EFS | EB | ES | SES | BBE | UAS | ESR | SESR | BBER, NEND | FEND

---

## SCPI Command Reference

### Performance Monitoring

---

:FETCh[1..n]:DATA:TELecom:SONet:LINE:PM:STATistics?

<b>Parameter(s)</b>	<b>Standard:</b> The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are: G829ISM   M2101ISM Selects the performance monitoring standard number. G829ISM: G.829 ISM M2101ISM: M.2101 ISM <b>Type:</b> The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are: EFS   EB   ES   SES   BBE   UAS   ESR   SESR   BBER Selects the performance monitoring statistics. EFS: Error Free Seconds (EFS). EB: Errored Block (EB). ES: Errored Seconds (ES). SES: Severely Errored Seconds (SES). BBE: Background Block Error (SES). UAS: Unavailable Second (UAS). ESR: Errored Second Ratio (ESR). SESR: Severely Errored Second Ratio (SESR). BBER: Background Block Error Ratio (BBER). <b>End:</b> The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed elements for this parameter are: NEND   FEND Selects Near-End or Far-End. NEND: the standard for Near-End. FEND: the standard for Far-End.
<b>Response Syntax</b>	<Statistics>

---

---

**:FETCh[1..n]:DATA:TELEcom:SONet:LINE:PM:STATistics?**

---

<b>Response(s)</b>	<b>Statistics:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the performance monitoring statistics of Line.
<b>Example(s)</b>	FETC:DATA:TEL:SON:LINE:PM:STAT? M2101ISM,ES,NEND
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PATtern:PM:STATistics?

---

## SCPI Command Reference

### *Performance Monitoring*

---

---

#### **:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:PM:STATistics?**

---

<b>Description</b>	<p>This query returns the performance monitoring statistics of HOP.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN -SONET /SDH BERT &gt; Result &gt; Performance Monitoring &gt; STS &gt; OTN SONET/SDH BERT &gt; Result &gt; Performance Monitoring &gt; AU</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:PM:STATistics? &lt;wsp&gt;G828ISM   M2101ISM, EFS   EB   ES   SES   BBE   UAS   ESR   SESR   BBER   SEP   SEPI, NEND   FEND</pre>

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:PM:STATistics?**

**Parameter(s)**

**Standard:**

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: G828ISM | M2101ISM

Selects the performance monitoring standard number.

G828ISM: G.828 ISM

M2101ISM: M.2101 ISM

**Type:**

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: EFS | EB | ES | SES | BBE | UAS | ESR | SESR | BBER | SEP | SEPI

Selects the performance monitoring statistics.

EFS: Error Free Seconds (EFS).

EB: Errored Block (EB).

ES: Errored Seconds (ES).

SES: Severely Errored Seconds (SES).

BBE: Background Block Error (SES).

UAS: Unavailable Second (UAS).

ESR: Errored Second Ratio (ESR).

SESR: Severely Errored Second Ratio (SESR).

BBER: Background Block Error Ratio (BBER).

SEP: Severely Errored Period (SEP).

SEPI: Severely Errored Period Intensity (SEPI).

**End:**

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: NEND | FEND

Selects Near-End or Far-End.

NEND: the standard for Near-End.

FEND: the standard for Far-End.

## SCPI Command Reference

### *Performance Monitoring*

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:PM:STATistics?**

**Response Syntax** <Statistics>

**Response(s)** **Statistics:**

The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element.

Returns the performance monitoring statistics of HOP.

**Example(s)** FETC:DATA:TEL:SDHS:HOP:PM:STAT? G828ISM,EFS,NEND

**See Also** FETCh[1..n]:DATA:TEL:SDHSonet:LINE:PM:STATistics?

---

## Traffic - Ethernet

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:LINE:UTILization?**

---

<b>Description</b>	<p>This query returns the utilization for TX/RX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Ethernet</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:LINE:UTILization? <wsp>RX   TX
<b>Parameter(s)</b>	<p><b>FTYP:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RX   TX</p> <p>Sets the transmitting/receiving line utilization.</p> <p>RX: sets the total of all receiving line utilization.</p> <p>TX: sets the total of all transmitting line utilization.</p>
<b>Response Syntax</b>	<Utilization>
<b>Response(s)</b>	<p><b>Utilization:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the transmitting/receiving line utilization.</p> <p>RX: returns the total of all receiving line utilization.</p> <p>TX: returns the total of all transmitting line utilization.</p>
<b>Example(s)</b>	SENS:DATA:TEL:ETH:PACK:LINE:UTIL? RX
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAME:RX?

---

## SCPI Command Reference

### Traffic - Ethernet

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:BANDwidth?

---

<b>Description</b>	<p>This query returns the Bandwidth for TX/RX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Ethernet</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:BANDwidth? <wsp>RX   TX
<b>Parameter(s)</b>	<p><b>FTYP:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RX   TX</p> <p>Selects the transmitting/receiving bandwidth.</p> <p>RX: sets the total of all receiving bandwidth.</p> <p>TX: sets the total of all transmitting bandwidth.</p>
<b>Response Syntax</b>	<Bandwidth>
<b>Response(s)</b>	<p><b>Bandwidth:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the transmitting/receiving frame bandwidth.</p> <p>RX: returns the total of all receiving frame bandwidth.</p> <p>TX: returns the total of all transmitting frame bandwidth.</p>
<b>Example(s)</b>	SENS:DATA:TEL:ETH:PACK:BAND? RX
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:LINE:UTILization?

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:FRAMe:RATE?**

<b>Description</b>	<p>This query returns the Frame Rate for TX/RX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary &gt; Traffic</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Ethernet</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:FRAMe:RATE? <wsp>RX   TX
<b>Parameter(s)</b>	<p><b>FTYP:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RX   TX</p> <p>Selects the transmitted/received number of frames.</p> <p>RX: sets the total of all received frames.</p> <p>TX: sets the total of all transmitted frames.</p>
<b>Response Syntax</b>	<Frame Rate>
<b>Response(s)</b>	<p><b>Frame Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the transmitting/receiving line utilization.</p> <p>RX: returns the total of all receiving line utilization.</p> <p>TX: returns the total of all transmitting line utilization.</p>
<b>Example(s)</b>	SENS:DATA:TEL:ETH:PACK:FRAM:RATE? RX
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:BANDwidth?

## SCPI Command Reference

### Traffic - Ethernet

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe:COUNt:RX?**

<b>Description</b>	<p>This query returns the number of received frames.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Ethernet &gt; RX Count</p> <p>Navigation Path: Test &gt; Results &gt; Summary &gt; Traffic &gt; RX Count</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe:COUNt:RX? &lt;wsp&gt;MULTicast   BROadcast   UNICast   NUNICast   FTOTal</p>
<b>Parameter(s)</b>	<p><b>Ftype:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MULTicast   BROadcast   UNICast   NUNICast   FTOTal</p> <p>Selects the frame type.</p> <p>MULTicast: Multicast</p> <p>BROadcast: Broadcast</p> <p>UNICast: Unicast</p> <p>NUNICast: Non-Unicast</p> <p>FTOTal: Total</p>
<b>Response Syntax</b>	<p>&lt;Count&gt;</p>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of received frames.</p> <p>MULTicast, selects the frame type as Multicast.</p> <p>BROadcast, selects the frame type as Broadcast.</p> <p>UNICast, selects the frame type as Unicast.</p> <p>NUNICast, selects the frame type as Non-Unicast.</p> <p>FTOTal, selects the frame type as Total.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FRAM:COUN:RX? MULT</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:FRAMe:COUNt?</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:FRAME:COUNt:TX?**

<b>Description</b>	<p>This query returns the number of transmitted frames.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Ethernet &gt; TX Count</p> <p>Navigation Path: Test &gt; Results &gt; Summary &gt; Traffic &gt; TX Count</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:FRAME:COUNt:TX? <wsp>MULTicast   BROadcast   UNICast   NUNICast   FTOtal
<b>Parameter(s)</b>	<p><b>Ftype:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MULTicast   BROadcast   UNICast   NUNICast   FTOtal</p> <p>Selects the frame type.</p> <p>MULTicast: Multicast</p> <p>BROadcast: Broadcast</p> <p>UNICast: Unicast</p> <p>NUNICast: Non-Unicast</p> <p>FTOtal: Total</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of transmitted frames.</p> <p>MULTicast, selects Multicast as the frame type.</p> <p>BROadcast, selects Broadcast as the frame type.</p> <p>UNICast, selects Unicast as the frame type.</p> <p>NUNICast, selects Non-Unicast as the frame type.</p> <p>FTOtal, selects Total as the frame type.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:FRAM:COUN:TX? MULT
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:FRAME:COUNt:RX?

## SCPI Command Reference

### Traffic - Ethernet

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZe:COUNT?**

<b>Description</b>	<p>This query returns the total number of all valid and invalid frames received.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Ethernet &gt; Frame Size &gt; RX Count</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZe:COUNT? &lt;wsp&gt;FLESS64   F64   F65TO127   F128TO255   F256TO511   F512TO1023   F1024TO1518   FMORE1518   FSTotal</p>
<b>Parameter(s)</b>	<p><b>Ftype:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FLESS64   F64   F65TO127   F128TO255   F256TO511   F512TO1023   F1024TO1518   FMORE1518   FSTotal</p> <p>Selects the frame size.</p> <p>FLESS64: frame size less than 64.</p> <p>F64: frame size equal to 64.</p> <p>F65TO127: frame size between 65 to 127.</p> <p>F128TO255: frame size between 128 to 255.</p> <p>F256TO511: frame size between 256 to 511.</p> <p>F512TO1023: frame size between 512 to 1023.</p> <p>F1024TO1518: frame size between 1024 to 1518.</p> <p>FMORE1518: frame size more than 1518.</p> <p>FSTotal: total frame size.</p>
<b>Response Syntax</b>	<p>&lt;Count&gt;</p>

---

---

**:SENSe[1..n]:DATA:TELecom:ETHernet:FSIZe:COUNT?**

<b>Response(s)</b>	<b>Count:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the total number of all valid and invalid frames received. FLESS64, selects frame size less than 64. F64, selects frame size equal to 64. F65TO127, selects frame size between 65 to 127. F128TO255, selects frame size between 128 to 255. F256TO511, selects frame size between 256 to 511. F512TO1023, selects frame size between 512 to 1023. F1024TO1518, selects frame size between 1024 to 1518. FMORE1518, selects frame size more than 1519. FSTotal, selects total frame size.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:FSIZ:COUNT? F64
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:ETHernet:PACKet:FRAMe:COUNT?

---

## SCPI Command Reference

### Traffic - Ethernet

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZe:PERCentage?

<b>Description</b>	<p>This query returns the percentage of each frames received.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Ethernet &gt; Frame Size &gt; %</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZe:PERCentage? &lt;wsp&gt;FLESS64   F64   F65TO127   F128TO255   F256TO511   F512TO1023   F1024TO1518   FMORE1518</code>
<b>Parameter(s)</b>	<p><b>Ftype:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FLESS64   F64   F65TO127   F128TO255   F256TO511   F512TO1023   F1024TO1518   FMORE1518</p> <p>Selects the frame size.</p> <p>FLESS64: frame size less than 64.</p> <p>F64: frame size equal to 64.</p> <p>F65TO127: frame size between 65 to 127.</p> <p>F128TO255: frame size between 128 to 255.</p> <p>F256TO511: frame size between 256 to 511.</p> <p>F512TO1023: frame size between 512 to 1023.</p> <p>F1024TO1518: frame size between 1024 to 1518.</p> <p>FMORE1518: frame size more than 1518.</p>
<b>Response Syntax</b>	<Percent>
<b>Response(s)</b>	<p><b>Percent:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the percentage of each frame received.</p>
<b>Example(s)</b>	<code>SENS:DATA:TEL:ETH:FSIZ:PERC? F64</code>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZe:COUNT?</code>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:FRAME:COUNT?**

<b>Description</b>	<p>This query returns the Frame count for TX/RX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Summary &gt; Traffic</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:PACKet:FRAME:COUNT? <wsp>RX   TX
<b>Parameter(s)</b>	<p><b>FTYP:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RX   TX</p> <p>Selects the number of frames transmitted.</p> <p>RX: Gives the total of all received valid and invalid frames.</p> <p>TX: Gives the total of all transmitted frames.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of transmitted or received frames.</p> <p>RXCount: Gives the total of all received valid and invalid frames.</p> <p>TXCount: Gives the total of all transmitted frames.</p>
<b>Example(s)</b>	SENS:DATA:TEL:ETH:PACK:FRAM:COUN? TX
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZE:COUNT?

---

## Traffic - Flow Control

---

:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES?

---

<b>Description</b>	This query returns the number of pause frames. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Traffic > Flow Control
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES?
<b>Response Syntax</b>	<Frames>
<b>Response(s)</b>	<b>Frames:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Reports the number of pause frames.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:PAUS:FRAM?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:ABORT?

---



---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:ABORT?**

---

<b>Description</b>	<p>This query returns the number of received pause frames with a Quanta equal to zero; cancelling the pause frames.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Flow Control</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:ABORT?
<b>Response Syntax</b>	<Abort>
<b>Response(s)</b>	<p><b>Abort:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of received pause frames.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:PAUS:FRAM:ABOR?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES?

---

## SCPI Command Reference

### Traffic - Flow Control

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:RX?

---

<b>Description</b>	This query reports total number of pause frames. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Traffic > Flow Control
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES:RX?
<b>Response Syntax</b>	<FrameRx>
<b>Response(s)</b>	<b>FrameRx:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the number of received flow control frames.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:PAUS:FRAM:RX?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:PAUSE:FRAMES?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSe:TIME**

---

<b>Description</b>	This command sets the total number of packet pause time for the link partner. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Traffic > Flow Control
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSe:TIME <wsp> <Pause Time>
<b>Parameter(s)</b>	<b>Pause Time:</b> The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the Pause packet time.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:PACK:PAUS:TIME 50
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:FRAME:COUNt:TX?

---

## SCPI Command Reference

### Traffic - Flow Control

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSe:TIME?**

<b>Description</b>	This query returns the total number of packet pause time received from the link partner. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Traffic > Flow Control
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSe:TIME?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<b>Time:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the total number of pause time received.
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:PACK:PAUS:TIME 50 SOUR:DATA:TEL:ETH:PACK:PAUS:TIME? Returns: 50
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:FRAMe:COUNt:TX?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSe:INJect**

---

<b>Description</b>	<p>This command injects pause time to the link partner.</p> <p>This command is an event and is not associated with any *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Flow Control</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSe:INJect
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:PACK:PAUS:INJ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJect

---

## SCPI Command Reference

### Traffic - Flow Control

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRESS**

<b>Description</b>	<p>This command sets the destination MAC address.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Flow Control &gt; Pause Injection &gt; Destination Mac Address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRESS &lt;wsp&gt; &lt;Set&gt;</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the destination MAC address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PINJ:MAC:DEST:ADDR 11:11:11:11:11:11</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRESS:ENAB le</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRess?**

<b>Description</b>	<p>This query returns the destination MAC address.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Flow Control &gt; Pause Injection &gt; Destination Mac Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRess?
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Destination MAC address value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PINJ:MAC:DEST:ADDR?</p> <p>Returns: "11:11:11:11:11:11"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRess:ENABle?

---

## SCPI Command Reference

### Traffic - Flow Control

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRess:ENABle**

<b>Description</b>	<p>This command enables the MAC destination address field.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Flow Control &gt; Pause Injection &gt; Destination Mac Address (checkbox)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRess:ENABle &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status of the destination MAC address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PINJ:MAC:DEST:ADDR:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:PINJ:MAC:DEST:ADDR:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRess</p>



**:SOURCE[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRess:ENABLE?**

<b>Description</b>	<p>This query returns the MAC destination address field.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Pause Injection &gt; Destination Mac Address (checkbox)</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRess:ENABLE?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Destination MAC address status.</p> <p>1, enables the destination MAC address.</p> <p>0, disables the destination MAC address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PINJ:MAC:DEST:ADDR:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:PINJ:MAC:DEST:ADDR:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:ETHernet:PINJection:MAC:DESTination:ADDRess?

---

## SCPI Command Reference

### Traffic - Flow Control

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSe:TIME:RX?

---

<b>Description</b>	<p>This query returns the total number of packet pause time received from the link partner.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Results &gt; Traffic &gt; Flow Control</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:PACKet:PAUSe:TIME:RX? &lt;wsp&gt;TOTALPAUSE   LASTPAUSE   MAXPAUSE   MINPAUSE</p>
<b>Parameter(s)</b>	<p><b>FTYP:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TOTALPAUSE   LASTPAUSE   MAXPAUSE   MINPAUSE</p> <p>Returns the pause time.</p> <p>TOTALPAUSE: returns the Total pause time.</p> <p>LASTPAUSE: sets the Last pause time.</p> <p>MINPAUSE: sets the Minimum pause time.</p> <p>MAXPAUSE: sets the Maximum pause time.</p>
<b>Response Syntax</b>	<p>&lt;Time&gt;</p>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the total number of pause time received.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:PACK:PAUS:TIME:RX? TOTALPAUSE</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:FRAME:COUNt:TX?</p>

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:PAUSe:FRAMes:TX?**

---

<b>Description</b>	This query reports total number of pause frames. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Results > Traffic > Flow Control
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:PAUSe:FRAMes:TX?
<b>Response Syntax</b>	<FrameRx>
<b>Response(s)</b>	<b>FrameRx:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Pause Time
<b>Example(s)</b>	FETC:DATA:TEL:ETH:PAUS:FRAM:TX?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:PAUSe:FRAMes?

---

## FTFL/PT

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?

<b>Description</b>	<p>This query returns the Fault Type Fault Location (FTFL) Fault Indication message to be generated.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; ODU &gt; FTFL</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication? <wsp>FORWARD   BACKward
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Indication>
<b>Response(s)</b>	<p><b>Indication:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Fault Type Fault Location (FTFL) Fault Indication message to be generated.</p> <p>NFAULT, No Fault (NFAULT) is selected as Fault Type Fault Location (FTFL) Fault Indication.</p> <p>SFAIL, Signal Fail (SFAIL) is selected as Fault Type Fault Location (FTFL) Fault Indication.</p> <p>SDEGRADE, Signal Degraded (SDEGRADE) is selected as Fault Type Fault Location (FTFL) Fault Indication.</p> <p>RESERVED, Reserved is selected as Fault Type Fault Location (FTFL) Fault Indication.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU1:FTFL:IND? FORW
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE?

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE?**

<b>Description</b>	<p>This query returns the Fault Type Fault Location (FTFL) Fault Indication code to be generated.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; ODU &gt; FTFL</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE? <wsp>FORWARD   BACKWARD
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKWARD</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKWARD, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the selection of the Fault Type Fault Location (FTFL) Fault Indication code to be generated.</p> <p>The values are from 0 to 255.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU1:FTFL:CODE? FORW
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?

## SCPI Command Reference

### FTFL/PT

---

---

#### :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier?

---

<b>Description</b>	<p>This query returns the Fault Type Fault Location (FTFL) Operator Identifier to be generated.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; ODU &gt; FTFL</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier? &lt;wsp&gt;FORWARD   BACKWARD</p>
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKWARD</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) Fault Type Fault Location (FTFL) to be generated.</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKWARD, sets the Backward configuration.</p>
<b>Response Syntax</b>	<p>&lt;Identifier&gt;</p>
<b>Response(s)</b>	<p><b>Identifier:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the value of the Fault Type Fault Location (FTFL) Operator Identifier (bytes 1 to 9 for both forward as well as backward) to be generated.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:OTN:ODU1:FTFL:IDEN? FORW</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication?</p>

---

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:FTFL:SPECific?**

<b>Description</b>	<p>This query returns the Fault Type Fault Location (FTFL) Operator Specific to be generated.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; ODU &gt; FTFL</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:FTFL:SPECific? <wsp>FORWARD   BACKward
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Operator Specific>
<b>Response(s)</b>	<p><b>Operator Specific:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the value of Fault Type Fault Location (FTFL) Operator Specific (bytes 118 for forward as well as backward) to be generated.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU1:FTFL:SPEC? FORWARD
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:FTFL:INDication?

## SCPI Command Reference

### FTFL/PT

---

---

#### :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication?

---

<b>Description</b>	<p>This query returns the Fault Type Fault Location (FTFL) Fault Indication message to be generated.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; OTU3e(1/2) &gt; FTFL</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; OTU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication? &lt;wsp&gt;FORWARD   BACKward</pre>
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Indication>
<b>Response(s)</b>	<p><b>Indication:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the FTFL Fault Indication message to be generated.</p> <p>NFAULT, No Fault (NFAULT) is selected as FTFL Fault Indication message.</p> <p>SFAIL, Signal Fail (SFAIL) is selected as FTFL Fault Indication message.</p> <p>SDEGRADE, Signal Degraded (SDEGRADE) is selected as FTFL Fault Indication message.</p> <p>RESERVED, Reserved is selected as FTFL Fault Indication message.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU3:E1:FTFL:IND? FORW
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE?

---



**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE?**

<b>Description</b>	<p>This query returns the Fault Type Fault Location (FTFL) Fault Indication code to be generated.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; OTU3e(1/2) &gt; FTFL</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; OTU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:CODE? <wsp>FORWARD   BACKward
<b>Parameter(s)</b>	<p><b>Fttl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKward</p> <p>Sets the configuration of the forward and backward ODU (Optical Channel Data Unit) FTFL (Fault Type Fault Location) to be generated.</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the FTFL Fault Indication Code to be generated.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU3:E1:FTFL:CODE? FORW
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication?

## SCPI Command Reference

### FTFL/PT

---

---

#### :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENTifier?

---

<b>Description</b>	<p>This query returns the FTFL Operator Identifier to be generated for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; OTU3e(1/2) &gt; FTFL</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; OTU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:IDENTifier? &lt;wsp&gt;FORWARD   BACKWARD</code>
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKWARD</p> <p>Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKWARD, sets the Backward configuration.</p>
<b>Response Syntax</b>	<code>&lt;Identifier&gt;</code>
<b>Response(s)</b>	<p><b>Identifier:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the value of the FTFL Operator Identifier (bytes 1 to 9 for forward, byte 129 to 137 for backward) to be generated.</p>
<b>Example(s)</b>	<code>SENS:DATA:TEL:OTN:ODU3:E1:FTFL:IDEN? FORW</code>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication?</code>

---

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:SPECific?**

<b>Description</b>	<p>This query returns the FTFL Operator Specific to be generated for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; OTU3e(1/2) &gt; FTFL</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; OTU(1/2)e &gt; FTFL</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:SPECific? <wsp>FORWARD   BACKWARD
<b>Parameter(s)</b>	<p><b>Ftfl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FORWARD   BACKWARD</p> <p>Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).</p> <p>FORWARD, sets the Forward configuration.</p> <p>BACKWARD, sets the Backward configuration.</p>
<b>Response Syntax</b>	<Specific>
<b>Response(s)</b>	<p><b>Specific:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the value of FTFL Operator Specific (bytes 10 to 127 for forward, byte 138 to 255 for backward) to be generated.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:ODU3:E1:FTFL:SPEC? FORW
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E[1..n]:FTFL:INDication?

## SCPI Command Reference

### FTFL/PT

---

#### :FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCODe:RECEived?

<b>Description</b>	<p>This query returns the received Payload Type as decimal code.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; ODU &gt; Payload Type</p> <p>In Multi-Channel OTN, when accessing an OPU LO, the channel must be set using SENS:DATA:TEL:OTN:OPU[1..n]:CHAN.</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCODe:RECEived?
<b>Response Syntax</b>	<Code>
<b>Response(s)</b>	<p><b>Code:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the received Payload Type Code.</p>
<b>Example(s)</b>	<p>OTN BERT:</p> <p>FETC:DATA:TEL:OTN:OPU1:PCOD:REC?</p> <p>For Multi-Channel OTN:</p> <p>SENS:DATA:TEL:OTN:OPU100:CHAN 2</p> <p>FETC:DATA:TEL:OTN:OPU100:PCOD:REC?</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:PTYPe:RECEived?</p> <p>SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:CHAN</p> <p>SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:ACHA</p>

---

**:FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:PTYPe:RECeived?**

<b>Description</b>	<p>This query returns the received Payload Type.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; ODU &gt; Payload Type</p> <p>In Multi-Channel OTN, when accessing an OPU LO, the channel must be set using SENS:DATA:TEL:OTN:OPU[1..n]:CHAN.</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:PTYPe:RECeived?
<b>Response Syntax</b>	<Payload Type>
<b>Response(s)</b>	<p><b>Payload Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the received Payload Type.</p>
<b>Example(s)</b>	<p>OTN BERT:</p> <p>FETC:DATA:TEL:OTN:OPU1:PTYP:REC?</p> <p>For Multi-Channel OTN:</p> <p>SENS:DATA:TEL:OTN:OPU100:CHAN 2</p> <p>FETC:DATA:TEL:OTN:OPU100:PTYP:REC?</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:PCODE:RECeived?</p> <p>SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:CHAN</p> <p>SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:ACHA</p>

---

## SCPI Command Reference

### FTFL/PT

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:COPY**

<b>Description</b>	<p>This command copies the received Payload Type to the expected Payload Type.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Results &gt; FTFL/PT &gt; ODU &gt; Payload Type</p> <p>NOTE: In Multi-Channel OTN, when copying an OPU LO Payload Type, the channel must be set using SENS:DATA:TEL:OTN:OPU[1..n]:CHAN. The Payload Type can also be copied on all channels by using SENS:DATA:TEL:OTN:OPU[1..n]:ACHA.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:COPY
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OPU1:COPY
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:TTI:SAPI:COPY SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:CHAN SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:ACHA

---

## OTL-SDT

---

**:FETCh[1..n]:DATA:TELEcom:OTL:SDT:STATistics?**

---

<b>Description</b>	<p>This query returns the STATISTICS FOR OTL.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;OTL-SDT&gt;Service disruption""</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTL:SDT:STATistics?
<b>Response Syntax</b>	<Statistics>
<b>Response(s)</b>	<p><b>Statistics:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the statistics.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SDT:STAT?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDT:SHORT?

---

## SCPI Command Reference

### OTL-SDT

---

#### :FETCh[1..n]:DATA:TELeom:OTL:SDT:LONGest?

<b>Description</b>	<p>This query returns the longest disruption time FOR OTL.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;OTL-SDT&gt;Service disruption""</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:OTL:SDT:LONGest? <wsp> <LaneNo>
<b>Parameter(s)</b>	<p><b>LaneNo:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the lane id</p>
<b>Response Syntax</b>	<Longest>
<b>Response(s)</b>	<p><b>Longest:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the longest disruption time for the selected lane id.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SDT:LONG? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:SDT:LONG?



**:FETCh[1..n]:DATA:TELeom:OTL:SDT:SHORtest?**

<b>Description</b>	<p>This query returns the shortest disruption time FOR OTL.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;OTL-SDT&gt;Service disruption""</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:OTL:SDT:SHORtest? <wsp> <LaneNo>
<b>Parameter(s)</b>	<p><b>LaneNo:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the lane id</p>
<b>Response Syntax</b>	<Shortest>
<b>Response(s)</b>	<p><b>Shortest:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the shortest disruption time for the selected lane id.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SDT:SHOR? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:OTL:SDT:LONG? 1

---

## SCPI Command Reference

### OTL-SDT

---

**:FETCh[1..n]:DATA:TELeom:OTL:SDT:LAST?**

<b>Description</b>	<p>This query returns the last disruption time FOR OTL.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;OTL-SDT&gt;Service disruption""</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELeom:OTL:SDT:LAST? &lt;wsp&gt; &lt;LaneNo&gt;</p>
<b>Parameter(s)</b>	<p><b>LaneNo:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the lane id</p>
<b>Response Syntax</b>	<p>&lt;Last&gt;</p>
<b>Response(s)</b>	<p><b>Last:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the last disruption time for the selected lane id.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTL:SDT:LAST? 1</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELeom:OTL:SDT:SHORT? 1</p>

**:FETCh[1..n]:DATA:TELeom:OTL:SDT:AVERAge?**

<b>Description</b>	<p>This query returns the average disruption time FOR OTL.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;OTL-SDT&gt;Service disruption""</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:OTL:SDT:AVERAge? <wsp> <LaneNo>
<b>Parameter(s)</b>	<p><b>LaneNo:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the lane id</p>
<b>Response Syntax</b>	<Average>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the average disruption time for the selected lane id.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SDT:AVER? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:OTL:SDT:LAST? 1

---

## SCPI Command Reference

### OTL-SDT

---

**:FETCh[1..n]:DATA:TELEcom:OTL:SDT:TOTal?**

<b>Description</b>	This query returns the total disruption time FOR OTL. At *RST condition, this value is device dependent. Navigation Path: Test > OTN BERT >Results>OTL-SDT>Service disruption""
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTL:SDT:TOTal? <wsp> <LaneNo>
<b>Parameter(s)</b>	<b>LaneNo:</b> The program data syntax for the first parameter is defined as a <numeric_value> element. Selects the lane id
<b>Response Syntax</b>	<Total>
<b>Response(s)</b>	<b>Total:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the total disruption time for the selected lane id.
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SDT:TOT? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTL:SDT:LONG? 1

**:FETCh[1..n]:DATA:TELeCom:OTL:SDT:COUNt?**

<b>Description</b>	<p>This query returns the disruption count FOR OTL.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; OTN BERT &gt;Results&gt;OTL-SDT&gt;Service disruption""</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeCom:OTL:SDT:COUNt? <wsp> <LaneNo>
<b>Parameter(s)</b>	<p><b>LaneNo:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the lane id</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of service disruption counts that happened since the beginning of the SDT test for the selected lane id.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SDT:COUN? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELeCom:OTL:SDT:LAST? 1

---

## SCPI Command Reference

### OTL-SDT

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTL:SDT:DEFect?

---

<b>Description</b>	<p>This query returns the layer on which service disruption time test is performed for OTN. At *RST condition, this value is device dependent. Navigation Path: Test &gt; OTN BERT &gt;Results&gt;OTL-SDT&gt;Service disruption&gt;Defect</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTL:SDT:DEFect?
<b>Response Syntax</b>	<Defect>
<b>Response(s)</b>	<p><b>Defect:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>This query returns the layer on which service disruption time test is performed for OTN for otl-sdt</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SDT:DEF?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDT:DEF?

---

<b>:FETCh[1..n]:DATA:TELeom:OTL:SDT:LONGest:DISRUption:DURation?</b>	
<b>Description</b>	This query returns the longest disruption duration. At *RST condition, this value is device dependent. Navigation Path: Test > OTN BERT >Results>OTL-SDT>Service disruption
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:OTL:SDT:LONGest:DISRUption:DURation?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<b>Time:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the longest disruption duration
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SDT:LONG:DISR:DUR?
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:OTL:SDT:LONG? 1

---

## SCPI Command Reference

### OTL-SDT

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTL:SDT:LONGest:DISRUption:LANE?

---

<b>Description</b>	This query returns the longest disruption lane. At *RST condition, this value is device dependent. Navigation Path: Test > OTN BERT >Results>OTL-SDT>Service disruption
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTL:SDT:LONGest:DISRUption:LANE?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<b>Time:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the longest disruption lane
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SDT:LONG:DISR:LANE?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTL:SDT:SHORT? 1

---



---

**:FETCh[1..n]:DATA:TELeom:OTL:SDT:LANE:DISRUption?**

---

<b>Description</b>	This query returns the lane disruption duration. At *RST condition, this value is device dependent. Navigation Path: Test > OTN BERT >Results>OTL-SDT>Service disruption
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:OTL:SDT:LANE:DISRUption?
<b>Response Syntax</b>	<Time>
<b>Response(s)</b>	<b>Time:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the lanes with disruption
<b>Example(s)</b>	FETC:DATA:TEL:OTL:SDT:LANE:DISR?
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:OTL:SDT:SHORT? 1

---

## GFP-F/GFP-T

**:FETCh[1..n]:DATA:TELecom:GFP:OVERview:COUNt:TX?**

<b>Description</b>	<p>This query returns the count of frames or payload bytes.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; GFP-F &gt; Frame Type &gt; TX &gt; Count</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Frame Type &gt; TX &gt; Count</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:GFP:OVERview:COUNt:TX? &lt;wsp&gt;DFRames   MFRames   IFRames   TFRames   DBYTESFORTX   MBYTESFORTX   IBYTESFORTX   TBYTESFORTX   DATBYTESFORTX</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames   IFRames   TFRames   DBYTESFORTX   MBYTESFORTX   IBYTESFORTX   TBYTESFORTX   DATBYTESFORTX</p> <p>Selects Count of the Trasmitted Frames and Bytes.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p>IFRames: IFRames</p> <p>TFRames: TFRames</p> <p>DBYTESFORTX: DBYTESFORTX</p> <p>MBYTESFORTX: MBYTESFORTX</p> <p>IBYTESFORTX: IBYTESFORTX</p> <p>TBYTESFORTX: TBYTESFORTX</p> <p>DATBYTESFORTX: DATBYTESFORTX</p>
<b>Response Syntax</b>	<p>&lt;Frames&gt;</p>

---

**:FETCh[1..n]:DATA:TELecom:GFP:OVERview:COUNT:TX?**

---

<b>Response(s)</b>	<b>Frames:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Count of Transmitted frames or payload bytes.
<b>Example(s)</b>	FETC:DATA:TEL:GFP:OVER:COUN:TX? DFR
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:GFP:OVERview:BANDwidth:TX?

---

## SCPI Command Reference

### GFP-F/GFP-T

---

#### :FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:TX?

<b>Description</b>	<p>This query returns the rate of frames or payload bytes.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; GFP-F &gt; Frame Type &gt; TX &gt; Rate(Unit/s)</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Frame Type &gt; TX &gt; Rate(Unit/s)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:TX? &lt;wsp&gt;DFRames   MFRames   IFRames   TFRames   DBYTESFORTX   MBYTESFORTX   IBYTESFORTX   TBYTESFORTX   DATBYTESFORTX</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames   IFRames   TFRames   DBYTESFORTX   MBYTESFORTX   IBYTESFORTX   TBYTESFORTX   DATBYTESFORTX</p> <p>Selects Rate of the Trasmitted Frames and Bytes.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p>IFRames: IFRames</p> <p>TFRames: TFRames</p> <p>DBYTESFORTX: DBYTESFORTX</p> <p>MBYTESFORTX: MBYTESFORTX</p> <p>IBYTESFORTX: IBYTESFORTX</p> <p>TBYTESFORTX: TBYTESFORTX</p> <p>DATBYTESFORTX: DATBYTESFORTX</p>
<b>Response Syntax</b>	<p>&lt;Frames&gt;</p>

---

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:TX?**

---

<b>Response(s)</b>	<b>Frames:</b> The response data syntax for this parameter is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the Rate of Transmitted frames or payload bytes.
<b>Example(s)</b>	FETC:DATA:TEL:GFP:OVER:RATE:TX? DFR
<b>See Also</b>	FETCH[1..n]:DATA:TELEcom:GFP:OVERview:EFFiciency:RX?

---

## SCPI Command Reference

### GFP-F/GFP-T

---

---

#### :FETCh[1..n]:DATA:TELEcom:GFP:OVERview:BANDwith:TX?

---

<b>Description</b>	<p>This query returns the Bandwidth usage of frames or payload bytes.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; GFP-F &gt; Transport Layer &gt; Bandwidth Usage(%) &gt; TX</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Transport Layer &gt; Bandwidth Usage(%) &gt; TX</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:BANDwith:TX?
<b>Response Syntax</b>	<TX>
<b>Response(s)</b>	<p><b>TX:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Bandwidth Percentage of Transmitted frames or payload bytes.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:OVER:BAND:TX?
<b>See Also</b>	FETCH[1..n]:DATA:TELEcom:GFP:OVERview:EFFiciency:RX?

---

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:EFFiciency:TX?**

<b>Description</b>	<p>This query returns the Mapping Efficiency of frames or payload bytes.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; GFP-F &gt; Transport Layer &gt; Mapping Efficiency (%) &gt; TX</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Transport Layer &gt; Mapping Efficiency (%) &gt; TX</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:EFFiciency:TX?
<b>Response Syntax</b>	<TX>
<b>Response(s)</b>	<p><b>TX:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Mapping Efficiency Percentage of Transmitted frames or payload bytes.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:OVER:EFF:TX?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:OVERview:BANDwidth:RX?

---

## SCPI Command Reference

### GFP-F/GFP-T

---

---

#### :FETCh[1..n]:DATA:TELeom:GFP:OVERview:COUNT:RX?

---

<b>Description</b>	<p>This query returns the count of frames or payload bytes.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; GFP-F &gt; Frame Type &gt; RX &gt; Count</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Frame Type &gt; RX &gt; Count</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELeom:GFP:OVERview:COUNT:RX? &lt;wsp&gt;DFRames   MFRames   IFRames   TFRames   RPTiframes   RPLiframes   INVFrames   DISFrames   DBYTESFORRX   MBYTESFORRX   IBYTESFORRX   TBYTESFORRX   RPTBYTESFORRX   DATBYTESFORRX</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames   IFRames   TFRames   RPTiframes   RPLiframes   INVFrames   DISFrames   DBYTESFORRX   MBYTESFORRX   IBYTESFORRX   TBYTESFORRX   RPTBYTESFORRX   DATBYTESFORRX</p> <p>Selects Count of the Received Frames and Bytes.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p>IFRames: IFRames</p> <p>TFRames: TFRames</p> <p>RPTiframes: RPTiframes</p> <p>RPLiframes: RPLiframes</p> <p>INVFrames: INVFrames</p> <p>DISFrames: DISFrames</p> <p>DBYTESFORRX: DBYTESFORRX</p> <p>MBYTESFORRX: MBYTESFORRX</p> <p>IBYTESFORRX: IBYTESFORRX</p> <p>TBYTESFORRX: TBYTESFORRX</p> <p>RPTBYTESFORRX: RPTBYTESFORRX</p> <p>DATBYTESFORRX: DATBYTESFORRX</p>

---



---

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:COUNT:RX?**

---

**Response Syntax** <Frames>**Response(s)** **Frames:**

The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element.

Returns the Count of Received frames or payload bytes.

**Example(s)** FETC:DATA:TEL:GFP:OVER:COUN:RX? DFR**See Also** FETCH[1..n]:DATA:TELEcom:GFP:OVERview:EFFiciency:TX?

## SCPI Command Reference

### GFP-F/GFP-T

---

---

#### :FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:RX?

---

<b>Description</b>	<p>This query returns the rate of frames or payload bytes.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; GFP-F &gt; Frame Type &gt; RX &gt; Rate(Unit/s)</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Frame Type &gt; RX &gt; Rate(Unit/s)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:RX? &lt;wsp&gt;DFRames   MFRames   IFRames   TFRames   RPTiframes   RPLiframes   INVFrames   DISFrames   DBYTESFORRX   MBYTESFORRX   IBYTESFORRX   TBYTESFORRX   RPTBYTESFORRX   DATBYTESFORRX</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames   IFRames   TFRames   RPTiframes   RPLiframes   INVFrames   DISFrames   DBYTESFORRX   MBYTESFORRX   IBYTESFORRX   TBYTESFORRX   RPTBYTESFORRX   DATBYTESFORRX</p> <p>Selects Rate of the Received Frames and Bytes.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p>IFRames: IFRames</p> <p>TFRames: TFRames</p> <p>RPTiframes: RPTiframes</p> <p>RPLiframes: RPLiframes</p> <p>INVFrames: INVFrames</p> <p>DISFrames: DISFrames</p> <p>DBYTESFORRX: DBYTESFORRX</p> <p>MBYTESFORRX: MBYTESFORRX</p> <p>IBYTESFORRX: IBYTESFORRX</p> <p>TBYTESFORRX: TBYTESFORRX</p> <p>RPTBYTESFORRX: RPTBYTESFORRX</p> <p>DATBYTESFORRX: DATBYTESFORRX</p>

---

---

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:RX?**

---

**Response Syntax** <Frames>**Response(s)** **Frames:**

The response data syntax for this parameter is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the Rate of Received frames or payload bytes.

**Example(s)** FETC:DATA:TEL:GFP:OVER:RATE:RX? DFR**See Also** FETCh[1..n]:DATA:TELEcom:GFP:OVERview:COUNt:TX?

## SCPI Command Reference

### GFP-F/GFP-T

---

---

#### :FETCh[1..n]:DATA:TELEcom:GFP:OVERview:BANDwith:RX?

---

<b>Description</b>	<p>This query returns the Bandwidth usage of frames or payload bytes.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; GFP-F &gt; Transport Layer &gt; Bandwidth Usage(%) &gt; RX</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Transport Layer &gt; Bandwidth Usage(%) &gt; RX</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:BANDwith:RX?
<b>Response Syntax</b>	<RX>
<b>Response(s)</b>	<p><b>RX:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Bandwidth Percentage of Received frames or payload bytes.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:OVER:BAND:RX?
<b>See Also</b>	FETCH[1..n]:DATA:TELEcom:GFP:OVERview:EFFiciency:TX?

---

**:FETCh[1..n]:DATA:TELecom:GFP:OVERview:EFFiciency:RX?**

<b>Description</b>	<p>This query returns the Mapping Efficiency of frames or payload bytes.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; GFP-F &gt; Transport Layer &gt; Mapping Efficiency (%) &gt; RX</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Transport Layer &gt; Mapping Efficiency (%) &gt; RX</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:GFP:OVERview:EFFiciency:RX?
<b>Response Syntax</b>	<RX>
<b>Response(s)</b>	<p><b>RX:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Mapping Efficiency Percentage of Received frames or payload bytes.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:OVER:EFF:RX?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:GFP:OVERview:RATE:TX?

---

## SCPI Command Reference

### GFP-F/GFP-T

---

---

#### :FETCh[1..n]:DATA:TELeom:GFP:SUPerblock:COUNT:TX?

---

<b>Description</b>	<p>This query returns the count of frames or payload bytes for superblock.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Superblock &gt; TX &gt; Count</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELeom:GFP:SUPerblock:COUNT:TX? &lt;wsp&gt;SUPERBLOCKTOTAL   SUPERBLOCKVALID</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SUPERBLOCKTOTAL   SUPERBLOCKVALID</p> <p>Selects type of the Transmitted Frames and Bytes for superblock.</p> <p>SuperblockValid: SuperblockValid superblockTotal: superblockTotal</p>
<b>Response Syntax</b>	<p>&lt;Frames&gt;</p>
<b>Response(s)</b>	<p><b>Frames:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Count of Transmitted frames or payload bytes.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:GFP:SUP:COUN:TX? SUPERBLOCKTOTAL</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELeom:GFP:OVERview:COUNT:RX?</p>

---

**:FETCh[1..n]:DATA:TELEcom:GFP:SUPerblock:RATE:TX?**

<b>Description</b>	<p>This query returns the rate of frames or payload bytes for superblock.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Superblock &gt; TX &gt; Rate(Unit/s)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:SUPerblock:RATE:TX? <wsp>SUPERBLOCKTOTAL   SUPERBLOCKVALID
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SUPERBLOCKTOTAL   SUPERBLOCKVALID</p> <p>Selects type of the Transmitted Frames and Bytes for superblock.</p> <p>SuperblockValid: SuperblockValid</p> <p>superblockTotal: SuperblockTotal</p>
<b>Response Syntax</b>	<Frames>
<b>Response(s)</b>	<p><b>Frames:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Rate of Transmitted frames or payload bytes.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:SUP:RATE:TX? SUPERBLOCKTOTAL
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:RX?

## SCPI Command Reference

### GFP-F/GFP-T

---

---

#### :FETCh[1..n]:DATA:TELEcom:GFP:SUPerblock:COUNT:RX?

---

<b>Description</b>	<p>This query returns the count of frames or payload bytes for superblock.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Superblock &gt; RX &gt; Count</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:GFP:SUPerblock:COUNT:RX? &lt;wsp&gt;SUPERBLOCKTOTAL   SUPERBLOCKVALID   SUPERBLOCKINVALID</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SUPERBLOCKTOTAL   SUPERBLOCKVALID   SUPERBLOCKINVALID</p> <p>Selects type of the Received Frames and Bytes for superblock.</p> <p>SuperblockValid: SuperblockValid</p> <p>SuperblockInvalid: SuperblockInvalid</p> <p>superblockTotal: SuperblockTotal</p>
<b>Response Syntax</b>	<p>&lt;Frames&gt;</p>
<b>Response(s)</b>	<p><b>Frames:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Count of Received frames or payload bytes.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:GFP:SUP:COUN:RX? SuperblockTotal</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:GFP:OVERview:COUNT:TX?</p>

---



**:FETCh[1..n]:DATA:TELEcom:GFP:SUPerblock:RATE:RX?**

<b>Description</b>	<p>This query returns the rate of frames or payload bytes for superblock.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; Superblock &gt; RX &gt; Rate(Unit/s)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:GFP:SUPerblock:RATE:RX?          &lt;wsp&gt;SUPERBLOCKTOTAL   SUPERBLOCKVALID   SUPERBLOCKINVALID</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SUPERBLOCKTOTAL   SUPERBLOCKVALID   SUPERBLOCKINVALID</p> <p>Selects type of the Received Frames and Bytes for superblock.</p> <p>SuperblockValid: SuperblockValid          SuperblockInvalid: SuperblockInvalid          superblockTotal: SuperblockTotal</p>
<b>Response Syntax</b>	<Frames>
<b>Response(s)</b>	<p><b>Frames:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Rate of Received frames or payload bytes.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:SUP:RATE:RX? SuperblockTotal
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:OVERview:RATE:TX?

## SCPI Command Reference

### GFP-F/GFP-T

---

#### :SENSe[1..n]:DATA:TELEcom:GFP:FRAMe:MISMATCH:COUNT?

<b>Description</b>	<p>This query returns the number of frames with fields not matching the Expected identifier, Channel Identifier and User Payload Identifier.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; GFP-F &gt; RX Mismatch</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; RX Mismatch</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:GFP:FRAMe:MISMATCH:COUNT? <wsp>UPIMIS   EXIMIS   RXMIS
<b>Parameter(s)</b>	<p><b>Value:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: UPIMIS   EXIMIS   RXMIS</p> <p>Selects count of frames for field not matching the expected identifier, Channel Identifier and User Payload Identifier</p>
<b>Response Syntax</b>	<COUNT>
<b>Response(s)</b>	<p><b>COUNT:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns count of frames for field not matching the expected identifier, Channel Identifier and User Payload Identifier</p>
<b>Example(s)</b>	SENS:DATA:TEL:GFP:FRAM:MISM:COUN? EXIMIS
<b>See Also</b>	SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTinuous?

---

**:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel:MISMATCH:COUNT?**

<b>Description</b>	<p>This query returns the number of frames with fields not matching the Payload Frame Check Sequence identifier.</p> <p>At *RST condition, this value is device-dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Results &gt; GFP-F &gt; RX Mismatch</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Results &gt; GFP-T &gt; RX Mismatch</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel:MISMATCH:COUNT? <wsp>PFIMIS
<b>Parameter(s)</b>	<p><b>Value:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PFIMIS</p> <p>Selects count of frames for field not matching the Payload Frame Check Sequence identifier</p>
<b>Response Syntax</b>	<COUNT>
<b>Response(s)</b>	<p><b>COUNT:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns count of frames for field not matching the Payload FCS Indicator.</p>
<b>Example(s)</b>	FETC:DATA:TEL:GFP:CHAN:MISM:COUN? PFIMIS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:GFP:ERRor:CHANnel:RATE?

---

## Streams - Throughput

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:MAXimum?**

<b>Description</b>	<p>This query returns the maximum throughput.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Throughput &gt;Maximum</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:MAXimum? <wsp><Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Maximum>
<b>Response(s)</b>	<p><b>Maximum:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the maximum throughput</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:THROughput:MAXimum? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MINimum?

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:MINimum?**

<b>Description</b>	<p>This query returns the minimum throughput.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Throughput &gt;Minimum</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:MINimum? <wsp><Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Minimum>
<b>Response(s)</b>	<p><b>Minimum:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the minimum throughput.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:THROughput:MINimum? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MAXimum?

## SCPI Command Reference

### Streams - Throughput

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THRoughput:CURRent?

---

<b>Description</b>	<p>This query returns the Current throughput.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Throughput &gt;Current</p>
<b>Syntax</b>	<pre>:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THRoughput:CURRent? &lt;wsp&gt; &lt;Tgen&gt;[, LTOR   RTOL]</pre>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<pre>&lt;Current&gt;</pre>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current throughput.</p>
<b>Example(s)</b>	<pre>FETCh:DATA:TELEcom:ETHernet:STReam:THRoughput:CURRent? 1</pre>
<b>See Also</b>	<pre>FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:AVERage?</pre>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THRoughput:AVERAge?**

<b>Description</b>	<p>This query returns the average throughput.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Throughput &gt;Average</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THRoughput:AVERAge? <wsp> <Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Average>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the average throughput.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:THRoughput:AVERAge? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:CURRent?

## SCPI Command Reference

### *Streams - Throughput*

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:CURRent:TOTal:RXRate?**

<b>Description</b>	This query return the Total RX Rate % received. At *RST condition, this value is device dependent. Navigation Path:Traffic Gen & Mon> Result >Stream >Throughput >Total RX Rate %.
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:CURRent:TOTal:RXRate?
<b>Response Syntax</b>	<Total Rx Rate>
<b>Response(s)</b>	<b>Total Rx Rate:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Total Rx rate value
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:THROughput:CURRent:TOTal:RXRate?
<b>See Also</b>	SOURce[1..n]:DATA:TEL:ETH:STR:PROFile:RATE?

---



**:FETCh[1..n]:DATA:TELecom:ETHernet:STReam:THRoughput:AVERAge:VERDict?**

<b>Description</b>	<p>This query returns the average throughput verdict status.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Traffic Gen &amp; Mon &gt; Results &gt; Streams &gt; Throughput &gt; Average</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:ETHernet:STReam:THRoughput:AVERAge:VERDict? <wsp> <Stream>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<VERDICT Status>
<b>Response(s)</b>	<p><b>VERDICT Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Average Throughput Verdict Status</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:STR:THR:AVER:VERD? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:STReam:OOSequence:VERDict? 1, COUNT

## SCPI Command Reference

### *Streams - Throughput*

---

**:FETCh[1..n]:DATA:TELeom:ETHernet:STReam:THROUGHput:CURRent:VERDict?**

<b>Description</b>	<p>This query returns the current throughput verdict status.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon &gt; Results&gt; Streams &gt; Throughput &gt;Current</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELeom:ETHernet:STReam:THROUGHput:CURRent:VERDict? &lt;wsp&gt; &lt;Stream&gt;</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p>
<b>Response Syntax</b>	<p>&lt;VERDICT Status&gt;</p>
<b>Response(s)</b>	<p><b>VERDICT Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Current Throughput Verdict Status.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:STR:THR:CURR:VERD? 1</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELeom:ETHernet:STReam:JITTer:MAXimum:VERDict? 1</p>

## Streams - Jitter

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MINimum?**

---

<b>Description</b>	<p>This query returns the minimum measured delay variations of the jitter.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Traffic Gen &amp; Mon &gt; Results &gt; Stream &gt; Jitter &gt; Minimum</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MINimum? <wsp> <Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Minimum>
<b>Response(s)</b>	<p><b>Minimum:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the minimum measured delay variations of jitter.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:JITTer:MINimum? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROUGHput:MAXimum?

---

## SCPI Command Reference

### Streams - Jitter

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MAXimum?

<b>Description</b>	<p>This query returns the maximum measured delay variations of the jitter.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Jitter &gt;Maximum</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:MAXimum? <wsp> <Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Maximum>
<b>Response(s)</b>	<p><b>Maximum:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the maximum measured delay variations of jitter.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:JITTer:MAXimum? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THRoughput:MINimum?

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:AVERage?**

<b>Description</b>	<p>This query returns the average of all valid measured delay variations of jitter.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Jitter &gt;Average</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:AVERage? <wsp> <Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Average>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the average of all valid measured delay variations of jitter.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:JITTer:AVERage? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THRoughput:CURRent?

## SCPI Command Reference

### Streams - Jitter

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:CURRent?

<b>Description</b>	<p>This query returns the current of all valid measured delay variations of jitter.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Jitter &gt;Current</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Summary &gt;Jitter &gt; Current</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:CURRent? <wsp> <Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the current of all valid measured delay variations of jitter.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:JITTer:CURRent? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROUGHput:AVERAge?

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:ESTimate?**

<b>Description</b>	<p>This query returns the estimated of all valid measured delay variations of jitter.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Jitter &gt; Estimated</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:JITTer:ESTimate? <wsp> <Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Estimated>
<b>Response(s)</b>	<p><b>Estimated:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the estimated of all valid measured delay variations of jitter.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:JITTer:EST? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THRoughput:MAXimum?

## SCPI Command Reference

### Streams - Jitter

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:STReam:JITTer:MAXimum:VERDict?**

<b>Description</b>	<p>This query returns the maximum Jitter Verdict status.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Traffic Gen &amp; Mon &gt; Results &gt; Streams &gt; Jitter &gt; Maximum</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:STReam:JITTer:MAXimum:VERDict? &lt;wsp&gt; &lt;Stream&gt;[, LTOR   RTOL]</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;VERDICT Status&gt;</p>
<b>Response(s)</b>	<p><b>VERDICT Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the maximum Jitter Verdict Status</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:STR:JITT:MAX:VERD? 1</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:ETHernet:STReam:THRoughput:CURRent:VERDict? 1</p>



## Streams - Latency

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MAXimum?**

---

<b>Description</b>	<p>This query returns the maximum Round Trip Delay recorded.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Latency &gt;Maximum</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MAXimum? <wsp> <Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Maximum>
<b>Response(s)</b>	<p><b>Maximum:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the maximum latency</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:LATency:MAXimum? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:HISTory?

---

## SCPI Command Reference

### Streams - Latency

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MINimum?

---

<b>Description</b>	<p>This query returns the minimum Round Trip Delay recorded.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Latency &gt;Minimum</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MINimum? &lt;wsp&gt; &lt;Tgen&gt;[, LTOR   RTOL]</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;Minimum&gt;</p>
<b>Response(s)</b>	<p><b>Minimum:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the mimum latency</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:STReam:LATency:MINimum? 1</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:CURRent?</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:AVERAge?**

<b>Description</b>	<p>This query returns the average Round Trip Delay value.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Latency &gt;Average</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:AVERAge? <wsp> <Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Average>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the average latency</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:LATency:AVERAge? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:SEConds?

## SCPI Command Reference

### Streams - Latency

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:CURRent?

---

<b>Description</b>	<p>This query returns the total number of successful measurements.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Latency &gt;Current</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Summary &gt; Latency</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:CURRent? <wsp> <Tgen>[, LTOR   RTOL]
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<Current>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the current latency.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STReam:LATency:CURRent? 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:COUNT?

---

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:STReam:LATency:MAXimum:VERDict?**

<b>Description</b>	<p>This query returns the maximum Latency Verdict status.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Traffic Gen &amp; Mon &gt; Results &gt; Streams &gt; Latency &gt; Maximum</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:STReam:LATency:MAXimum:VERDict? &lt;wsp&gt; &lt;Stream&gt;[, LTOR   RTOL]</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;VERDICT Status&gt;</p>
<b>Response(s)</b>	<p><b>VERDICT Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Maximum Latency Verdict Status.</p> <p>PASS, verdict is Pass.</p> <p>FAIL, verdict is Fail.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:ETH:STR:LAT:MAX:VERD? 1</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELecom:ETHernet:STReam:FLOSS:VERDict? 1, RATE</p>

---

## Streams - Frame Loss / Out-of-Sequence

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:HISTory?**

<b>Description</b>	<p>This query returns the history status of stream errors analysis errors.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Frame Loss /Out-of-sequence</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:HISTory? <wsp> <Tgen>, OUTSequence   FLOs
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OUTSequence   FLOs</p> <p>Selects the type of stream errors analysis error.</p> <p>OUTSequence: Out of Sequence</p> <p>FLOs: Frame Loss</p>
<b>Response Syntax</b>	<History>
<b>Response(s)</b>	<p><b>History:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the history status of stream errors analysis error.</p> <p>PRESENT, indicates that the stream errors and analysis error has occurred.</p> <p>ABSENT, indicates that the stream errors and analysis error has not occurred.</p> <p>INACTIVE, indicates that the stream errors and analysis error has not run yet.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ERRor:SANalyzer:HISTory? 1, FLOs
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:CURRent?**

---

<b>Description</b>	<p>This query returns the current status of stream errors analysis error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Frame Loss /Out-of-sequence</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:CURRent? &lt;wsp&gt; &lt;Tgen&gt;, OUTSequence   FLOs</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OUTSequence   FLOs</p> <p>Selects the type of stream errors analysis error.</p> <p>OUTSequence: Out of Sequence</p> <p>FLOs: Frame Loss</p>
<b>Response Syntax</b>	<p>&lt;Current&gt;</p>
<b>Response(s)</b>	<p><b>Current:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the current status of stream errors analysis error.</p> <p>PRESENT, indicates that the stream errors and analysis error has occurred.</p> <p>ABSENT, indicates that the stream errors and analysis error has not occurred.</p> <p>INACTIVE, indicates that the stream errors and analysis error has not run yet.</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ERRor:SANalyzer:CURRent? 1, FLOs</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:RATE?</p>

---

## SCPI Command Reference

### *Streams - Frame Loss / Out-of-Sequence*

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:SEConds?

---

<b>Description</b>	<p>This query returns the number of seconds within which stream errors analysis error occurred.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Frame Loss /Out-of-sequence</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:SEConds? &lt;wsp&gt; &lt;Tgen&gt;, OUTSequence   FLOs</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OUTSequence   FLOs</p> <p>Selects the type of stream errors analysis error.</p> <p>OUTSequence: Out of Sequence</p> <p>FLOs: Frame Loss</p>
<b>Response Syntax</b>	<p>&lt;Seconds&gt;</p>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of stream errors analysis error.</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ERRor:SANalyzer:SEConds? 1, FLOs</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:COUNT?</p>

---



---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:COUNT?**

---

<b>Description</b>	<p>This query returns the count for stream errors analysis error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Frame Loss /Out-of-sequence</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:COUNT? &lt;wsp&gt; &lt;Tgen&gt;, OUTSequence   FLOs</p>
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OUTSequence   FLOs</p> <p>Selects the type of stream errors analysis error.</p> <p>OUTSequence: Out of Sequence</p> <p>FLOs: Frame Loss</p>
<b>Response Syntax</b>	<p>&lt;Count&gt;</p>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of stream errors analysis error.</p>
<b>Example(s)</b>	<p>FETCh:DATA:TELEcom:ETHernet:ERRor:SANalyzer:COUNT? 1, FLOs</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:CURRent?</p>

---

## SCPI Command Reference

### Streams - Frame Loss / Out-of-Sequence

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:RATE?

---

<b>Description</b>	<p>This query returns the current rate of stream errors analysis error.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path:Traffic Gen &amp; Mon&gt; Results &gt;Stream &gt;Frame Loss /Out-of-sequence</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:SANalyzer:RATE? <wsp> <Tgen>, OUTSequence   FLOSs
<b>Parameter(s)</b>	<p><b>Tgen:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OUTSequence   FLOSs</p> <p>Selects the type of stream errors analysis error.</p> <p>OUTSequence: Out of Sequence</p> <p>FLOSs: Frame Loss</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the current rate of stream errors analysis error.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ERRor:SANalyzer:RATE? 1, FLOSs
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:EOTN:ERRor:PHYSical:HISTory?

---

<b>:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:OOSequence:VERDict?</b>	
<b>Description</b>	<p>This query returns the Out Of Sequence COUNT/RATE Verdict status.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Traffic Gen &amp; Mon &gt; Results &gt; Streams &gt; Frame Loss /Out Of Sequence &gt; Out Of Sequence</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:OOSequence:VERDict? &lt;wsp&gt; &lt;Stream&gt;, COUNT   RATE[, LTOR   RTOL]</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: COUNT   RATE</p> <p>Selects Frameless Type.</p> <p>COUNT: Count</p> <p>RATE: Rate</p> <p><b>Direction:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;VERDICT Status&gt;</p>

---

## SCPI Command Reference

### *Streams - Frame Loss / Out-of-Sequence*

---

<b>:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:OOSequence:VERDict?</b>	
<b>Response(s)</b>	<b>VERDICT Status:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Out Of Sequence Verdict Status. PASS, verdict is Pass. FAIL, verdict is Fail.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:STR:OOS:VERD? 1, COUNT
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:THROughput:AVERage:VERDict? 1

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:FLOsS:VERDICT?**

<b>Description</b>	<p>This query returns the Frame Loss COUNT/RATE Verdict status.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Traffic Gen &amp; Mon &gt; Results &gt; Streams &gt; Frame Loss /Out Of Sequence &gt; Frame Loss</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:FLOsS:VERDICT? &lt;wsp&gt; &lt;Stream&gt;, COUNT   RATE[, LTOR   RTOL]</p>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the stream from 1 to 16.</p> <p><b>Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: COUNT   RATE</p> <p>Selects Frameloss Type.</p> <p>COUNT: Count</p> <p>RATE: Rate</p> <p><b>Direction:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>This parameter is optional</p> <p>Selects the direction</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p>
<b>Response Syntax</b>	<p>&lt;VERDICT Status&gt;</p>

---

## SCPI Command Reference

### *Streams - Frame Loss / Out-of-Sequence*

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:FLOs:VERDict?**

<b>Response(s)</b>	<b>VERDICT Status:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Frame Loss Verdict Status. PASS, verdict is Pass. FAIL, verdict is Fail.
<b>Example(s)</b>	FETC:DATA:TEL:ETH:STR:FLOS:VERD? 1, RATE
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:LATency:MAXimum:VERDict? 1

---

## Service Configuration - Ramp

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSs:VERDict?**

---

**Description** This Query returns Ramp Test Frame Loss Verdict.  
Navigation Path: Test Setup > EtherSAM > Results > Service CONFig > Ramp > Committed Steps > Frame Loss(%)

**Syntax** :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSs:VERDict?  
<wsp> <Service>, LTOR | RTOL, 1 | 2 | 3 | 4 | 5 | 6 | 7 | CIR

---

## SCPI Command Reference

### *Service Configuration - Ramp*

---

**:FETCh[1..n]:DATA:TELEcom:ETHerNet:ESAM:SCOTest:RAMP:FLOs:VERDict?**

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Step:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 1   2   3   4   5   6   7   CIR</p> <p>Sets the Step.</p> <p>1: 1</p> <p>2: 2</p> <p>3: 3</p> <p>4: 4</p> <p>5: 5</p> <p>6: 6</p> <p>7: 7</p> <p>CIR: CIR</p>
<b>Response Syntax</b>	<Verdict>

---



---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOs:VERDict?**

---

**Response(s)****Verdict:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Verdict

PASS, verdict is Pass.

FAIL, verdict is Fail.

**Example(s)**

FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOs:VERDict? 1, LTOR, 1

**See Also**

SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:DEFault

---

## SCPI Command Reference

### Service Configuration - Ramp

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter:VERDict?

<b>Description</b>	<p>This Query returns Ramp Test Max Jitter Verdict.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Service CONFig&gt;Ramp&gt;Committed Steps&gt;Max Jitter(ms)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter:VERDict? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, 1   2   3   4   5   6   7   CIR</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Step:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 1   2   3   4   5   6   7   CIR</p> <p>Sets the Step.</p> <p>1: 1</p> <p>2: 2</p> <p>3: 3</p> <p>4: 4</p> <p>5: 5</p> <p>6: 6</p> <p>7: 7</p> <p>CIR: CIR</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter:VERDict?****Response Syntax** <Verdict>**Response(s)** **Verdict:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Verdict

PASS, verdict is Pass.

FAIL, verdict is Fail.

**Example(s)** FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter:VERDict? 1,  
LTOR,1**See Also** FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOs:VERDict?

## SCPI Command Reference

### Service Configuration - Ramp

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency:VERDict?

<b>Description</b>	<p>This Query returns Ramp Test Round Trip Latency Verdict.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Service CONFig&gt;Ramp&gt;Committed Steps&gt;Round Trip Latency(ms)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency:VERDict? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, 1   2   3   4   5   6   7   CIR</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Step:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 1   2   3   4   5   6   7   CIR</p> <p>Sets the Step.</p> <p>1: 1</p> <p>2: 2</p> <p>3: 3</p> <p>4: 4</p> <p>5: 5</p> <p>6: 6</p> <p>7: 7</p> <p>CIR: CIR</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency:VERDict?****Response Syntax** <Verdict>**Response(s)** **Verdict:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Verdict

PASS, verdict is Pass.

FAIL, verdict is Fail.

**Example(s)** FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency:VERDict? 1,  
LTOR,1**See Also** FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter:VERDict?

## SCPI Command Reference

### Service Configuration - Ramp

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate:VERDict?**

<b>Description</b>	<p>This Query returns Ramp Test Avg. Rx Verdict.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Service CONFig&gt;Ramp&gt;Committed Steps&gt;Average RX Rate</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate:VERDict? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, CIREIR   TRAFFICPOLicing</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Step:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CIREIR   TRAFFICPOLicing</p> <p>Sets the Step</p> <p>CIREIR: CIR+EIR</p> <p>TRAFFICPOLicing: Traffic Policing</p>
<b>Response Syntax</b>	<p>&lt;Verdict&gt;</p>

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate:VERDict?**

---

<b>Response(s)</b>	<b>Verdict:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Verdict PASS, verdict is Pass. FAIL, verdict is Fail.
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate:VERDict? 1, L TOR,CIREIR
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency:VERDict?

---

## Service Configuration - Burst

:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOSS?

<b>Description</b>	<p>This query returns Frame Loss statistics.</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Service CONFig &gt; Burst &gt; Committed(CBS) &gt; Frame Loss(%)</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Service CONFig &gt; Burst &gt; Excess(EBS) &gt; Frame Loss(%)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOSS? <wsp> <Service>, LTOR   RTOL, CBS   EBS
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS   EBS</p> <p>Sets the type of Burst.</p> <p>CBS</p> <p>EBS</p>
<b>Response Syntax</b>	<Frame Loss>



---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOs?**

---

<b>Response(s)</b>	<b>Frame Loss:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Frame Loss statistics
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOs? 1, LTOR,CBS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME?

---

## SCPI Command Reference

### Service Configuration - Burst

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter?**

<b>Description</b>	<p>This query returns Max Jitter statistics.</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Service CONFig &gt; Burst &gt; Committed(CBS) &gt; MAX Jitter(ms)</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Service CONFig &gt; Burst &gt; Excess(EBS) &gt; MAX Jitter(ms)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, CBS   EBS</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS   EBS</p> <p>Sets the type of Burst.</p> <p>CBS</p> <p>EBS</p>
<b>Response Syntax</b>	<p>&lt;Max Jitter&gt;</p>

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter?**

---

<b>Response(s)</b>	<b>Max Jitter:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Max Jitter statistics
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter? 1, LTOR,CBS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:DURation?

---

## SCPI Command Reference

### Service Configuration - Burst

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency?

<b>Description</b>	<p>This query returns Round Trip Latency statistics.</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Service CONFig&gt;Burst&gt;Committed(CBS)&gt;Round Trip Latency(ms)</p> <p>Navigation Path: Test Setup &gt;EtherSAM&gt;Results&gt;Service CONFig&gt;Burst&gt;Excess(EBS)&gt;Round Trip Latency(ms)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency? <wsp> <Service>, LTOR   RTOL, CBS   EBS
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS   EBS</p> <p>Sets the type of Burst.</p> <p>CBS</p> <p>EBS</p>
<b>Response Syntax</b>	<Round Trip Latency>

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency?**

---

<b>Response(s)</b>	<b>Round Trip Latency:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Round Trip Latency statistics
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency? 1, LTOR,CBS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETH:ESAM:BURSt:CBS:TIME?

---

## SCPI Command Reference

### Service Configuration - Burst

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate?**

<b>Description</b>	<p>This query returns Average RX Rate statistics.</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Service CONFig &gt; Burst &gt; Committed(CBS) &gt; Average RX Rate (%)</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Service CONFig &gt; Burst &gt; Excess(EBS) &gt; MAX RX Rate (%)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, CBS   EBS</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS   EBS</p> <p>Sets the type of Burst.</p> <p>CBS</p> <p>EBS</p>
<b>Response Syntax</b>	<p>&lt;Average RX Rate&gt;</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate?**

<b>Response(s)</b>	<b>Average RX Rate:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Average RX Rate statistics
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate? 1, LTOR,CBS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETH:ESAM:BURSt:EBS:TIME?

---

## SCPI Command Reference

### *Service Configuration - Burst*

---

---

#### **:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:TXRate?**

---

<b>Description</b>	This Query returns Ramp Test Tx rate. Navigation Path: Test Setup > EtherSAM > Results > Service CONFig > Ramp > committed Steps > TX Rate(%)
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:TXRate? <wsp> <Service>, LTOR   RTOL, 1   2   3   4   5   6   7   CIR   CIREIR   TRAFFICPOLicing

---



**:FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:SCOTest:RAMP:TXRate?****Parameter(s)****Service:**

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the Service number 1 or 10.

**Direction:**

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: LTOR | RTOL

Selects the Direction.

(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)

LTOR: Local to Remote

RTOL: Remote to Local

**Step:**

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: 1 | 2 | 3 | 4 | 5 | 6 | 7 | CIR | CIREIR | TRAFFICPOLicing

Sets the Step.

1: 1

2: 2

3: 3

4: 4

5: 5

6: 6

7: 7

CIR: CIR

CIREIR: CIREIR

TRAFFICPOLicing: TRAFFICPOLicing

**Response Syntax** <TX Rate>

## SCPI Command Reference

### *Service Configuration - Burst*

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:TXRate?**

<b>Response(s)</b>	<b>TX Rate:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Ramp Test Tx rate
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:TXRate? 1, LTOR,1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETH:ESAM:BURSt:TBURst:TIME?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSs?**

**Description** This Query returns Ramp Test Frame Loss Statistics.  
 Navigation Path: Test Setup > EtherSAM > Results > Service CONFig > Ramp > Committed Steps > Frame Loss(%)

**Syntax** :FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSs? <wsp>  
 <Service>, LTOR | RTOL, 1 | 2 | 3 | 4 | 5 | 6 | 7 | CIR | CIREIR | TRAFFICPOLicing

## SCPI Command Reference

### Service Configuration - Burst

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOsS?**

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Step:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 1   2   3   4   5   6   7   CIR   CIREIR   TRAFFICPOLicing</p> <p>Sets the Step.</p> <p>1: 1</p> <p>2: 2</p> <p>3: 3</p> <p>4: 4</p> <p>5: 5</p> <p>6: 6</p> <p>7: 7</p> <p>CIR: CIR</p> <p>CIREIR: CIREIR</p> <p>TRAFFICPOLicing: TRAFFICPOLicing</p>
<b>Response Syntax</b>	<Frame Loss>

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSs?**

<b>Response(s)</b>	<p><b>Frame Loss:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Ramp Test Frame Loss Statistics</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:FLOSs? 1, LTOR, 1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETH:ESAM:BURSt:TOT?

---

## SCPI Command Reference

### *Service Configuration - Burst*

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter?**

---

<b>Description</b>	This Query returns Ramp Test Max Jitter. Navigation Path: Test Setup > EtherSAM > Results > Service CONFig > Ramp > Committed Steps > Max Jitter(ms)
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter? <wsp> <Service>, LTOR   RTOL, 1   2   3   4   5   6   7   CIR   CIREIR   TRAFFICPOLicing

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter?**

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Step:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 1   2   3   4   5   6   7   CIR   CIREIR   TRAFFICPOLicing</p> <p>Sets the Step.</p> <p>1: 1</p> <p>2: 2</p> <p>3: 3</p> <p>4: 4</p> <p>5: 5</p> <p>6: 6</p> <p>7: 7</p> <p>CIR: CIR</p> <p>CIREIR: CIREIR</p> <p>TRAFFICPOLicing: TRAFFICPOLicing</p>
<b>Response Syntax</b>	<Max Jitter>

---

## SCPI Command Reference

### *Service Configuration - Burst*

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter?**

<b>Response(s)</b>	<b>Max Jitter:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Ramp Test Max Jitter
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:MAXJitter? 1, LTOR,1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETH:ESAM:GLOB:SPRTTest:DUR

---



---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency?**

---

<b>Description</b>	This Query returns Ramp Test Round Trip Latency. Navigation Path: Test Setup > EtherSAM > Results > Service CONFig > Ramp > Committed Steps > Round Trip Latency(ms)
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency? <wsp> <Service>, LTOR   RTOL, 1   2   3   4   5   6   7   CIR   CIREIR   TRAFFICPOLicing

---

## SCPI Command Reference

### Service Configuration - Burst

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:SCOTest:RAMP:RTLatency?**

<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Step:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 1   2   3   4   5   6   7   CIR   CIREIR   TRAFFICPOLicing</p> <p>Sets the Step.</p> <p>1: 1</p> <p>2: 2</p> <p>3: 3</p> <p>4: 4</p> <p>5: 5</p> <p>6: 6</p> <p>7: 7</p> <p>CIR: CIR</p> <p>CIREIR: CIREIR</p> <p>TRAFFICPOLicing: TRAFFICPOLicing</p>
<b>Response Syntax</b>	<Round Trip Latency>

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency?**

<b>Response(s)</b>	<b>Round Trip Latency:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Ramp Test Round Trip Latency
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:RTLatency? 1, LTOR,1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETH:ESAM:GLOB:SPRTest:DUR?

---

## SCPI Command Reference

### *Service Configuration - Burst*

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate?**

---

<b>Description</b>	This Query returns Ramp Test Avg. Rx Rate. Navigation Path: Test Setup >EtherSAM>Results>Service CONFig>Ramp>Committed Steps>Average RX Rate
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate? <wsp> <Service>, LTOR   RTOL, 1   2   3   4   5   6   7   CIR   CIREIR   TRAFFICPOLicing

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate?****Parameter(s)****Service:**

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the Service number 1 or 10.

**Direction:**

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: LTOR | RTOL

Selects the Direction.

(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)

LTOR: Local to Remote

RTOL: Remote to Local

**Step:**

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: 1 | 2 | 3 | 4 | 5 | 6 | 7 | CIR | CIREIR | TRAFFICPOLicing

Sets the Step.

1: 1

2: 2

3: 3

4: 4

5: 5

6: 6

7: 7

CIR: CIR

CIREIR: CIREIR

TRAFFICPOLicing: TRAFFICPOLicing

**Response Syntax** <Average RX Rate>

## SCPI Command Reference

### *Service Configuration - Burst*

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate?**

<b>Response(s)</b>	<b>Average RX Rate:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the Ramp Test Avg. Rx Rate
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:RAMP:ARXRate? 1, LTOR,1
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETH:ESAM:GLOB:TDURATION:EST?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOs:VERDict?**

<b>Description</b>	<p>This query returns Burst Test Frame Loss verdict.</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Service CONFig &gt; Burst &gt; Committed(CBS) &gt; Frame Loss(%)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOs:VERDict? <wsp> <Service>, LTOR   RTOL, CBS
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS</p> <p>Sets Burst test type as CBS.</p>
<b>Response Syntax</b>	<Verdict>

## SCPI Command Reference

### *Service Configuration - Burst*

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOs:VERDict?**

<b>Response(s)</b>	<b>Verdict:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Verdict PASS, verdict is Pass. FAIL, verdict is Fail.
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:FLOs:VERDict? 1, LTOR,CBS
<b>See Also</b>	FETC[1..n]:DATA:TELEcom:ETHernet:ESAM:TESTs:VERDict?

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter:VERDict?**

<b>Description</b>	<p>This query returns Burst Tst Max Jitter Verdict.</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Service CONFig&gt;Burst&gt;Committed(CBS)&gt;MAX Jitter(ms)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter:VERDict? &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, CBS</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS</p> <p>Sets Burst test type as CBS.</p>
<b>Response Syntax</b>	<p>&lt;Verdict&gt;</p>

## SCPI Command Reference

### *Service Configuration - Burst*

---

**:FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter:VERDict?**

<b>Response(s)</b>	<b>Verdict:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Verdict PASS, verdict is Pass. FAIL, verdict is Fail.
<b>Example(s)</b>	FETCh:DATA:TELecom:ETHernet:ESAM:SCOTest:BURSt:MAXJitter:VERDict? 1, L TOR,CBS
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:ETHernet:ESAM:SCOTest:VERDict?

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency:VERDict?**

<b>Description</b>	<p>This query returns Burst Test Round Trip Latency Verdict.</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Service CONFig &gt; Burst &gt; Committed(CBS) &gt; Round Trip Latency(ms)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency:VERDict? <wsp> <Service>, LTOR   RTOL, CBS
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: CBS</p> <p>Sets Burst test type as CBS.</p>
<b>Response Syntax</b>	<Verdict>

## SCPI Command Reference

### *Service Configuration - Burst*

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency:VERDict?**

<b>Response(s)</b>	<b>Verdict:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst Tst Max Jitter Verdict
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:RTLatency:VERDict? 1, LTOR,CBS
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SPRTest:VERDict? 1, LTOR

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate:VERDict?**

<b>Description</b>	<p>This query returns Average Rx Rate Verdict.</p> <p>Navigation Path: Test Setup &gt; EtherSAM &gt; Results &gt; Service          CONFIg&gt;Burst&gt;Committed(CBS)&gt;Average RX Rate (%)</p> <p>Navigation Path: Test Setup &gt; Ether SAM &gt; Results &gt; Service          CONFIg&gt;Burst&gt;Excess(EBS)&gt;MAX RX Rate (%)</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate:VERDict?          &lt;wsp&gt; &lt;Service&gt;, LTOR   RTOL, EBS</p>
<b>Parameter(s)</b>	<p><b>Service:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Service number 1 or 10.</p> <p><b>Direction:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LTOR   RTOL</p> <p>Selects the Direction.</p> <p>(For DTS used both LTOR,RTOL as direction and For Non DTS use LTOR as direction)</p> <p>LTOR: Local to Remote</p> <p>RTOL: Remote to Local</p> <p><b>Burst Size Type :</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: EBS</p> <p>Sets Burst test type as EBS.</p>
<b>Response Syntax</b>	<p>&lt;Verdict&gt;</p>

## SCPI Command Reference

### *Service Configuration - Burst*

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate:VERDict?**

<b>Response(s)</b>	<b>Verdict:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Verdict PASS, verdict is Pass. FAIL, verdict is Fail.
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:ESAM:SCOTest:BURSt:ARXRate:VERDict? 1, LTOR,EBS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ESAM:RAMP:STEP:TIME

---

## Labels

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel?**

---

<b>Description</b>	<p>This query returns the expected path signal label for the receiver of High Order Path (HOP). At *RST condition, the CONFig is set to a device-dependent value.</p> <p>Navigation Path: ""Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Test Configurator &gt; Results &gt; Traces &gt; Sonet/SDH &gt; Labels &gt; STS Path/AU Path &gt; Received ""</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel?
<b>Response Syntax</b>	<Label>
<b>Response(s)</b>	<p><b>Label:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the path signal label for the receiver.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:PATH:LAB?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:PATH:LABel

---

## MPLS

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:BANDwidth?**

<b>Description</b>	<p>This query returns the frame bandwidth of Multi Protocol Label Switching (MPLS) in megabits per second.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Results &gt; Streams &gt; MPLS &gt; Total RX MPLS &gt; Ethernet BW(%)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:BANDwidth?
<b>Response Syntax</b>	<Bandwidth>
<b>Response(s)</b>	<p><b>Bandwidth:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the ethernet bandwidth (RX).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:MPLS ON</p> <p>SENS:DATA:TEL:ETH:STR:MPLS:RX:BAND?</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:RATE?**

<b>Description</b>	<p>This query returns the Multi Protocol Label Switching (MPLS) frame rate in frame per second.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Results &gt; Streams &gt; MPLS &gt; Total RX MPLS &gt; Frame Rate (Frame/s)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:RATE?
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frame rate (RX).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:MPLS ON</p> <p>SENS:DATA:TEL:ETH:STR:MPLS:RX:RATE?</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL?

---

## SCPI Command Reference

### MPLS

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:UTILization?

---

<b>Description</b>	<p>This query returns the Multi Protocol Label Switching (MPLS) frame utilization in percentage.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Results &gt; Streams &gt; MPLS &gt; Total RX MPLS &gt; Line Utilization(%)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:UTILization?
<b>Response Syntax</b>	<Utilization>
<b>Response(s)</b>	<p><b>Utilization:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frame utilization (RX).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:MPLS ON</p> <p>SENS:DATA:TEL:ETH:STR:MPLS:RX:UTIL?</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:MPLS:HEADers

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:COUNT?**

<b>Description</b>	<p>This query returns total of all Multi Protocol Label Switching (MPLS) received valid and invalid frames.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Results &gt; Streams &gt; MPLS &gt; Total RX MPLS &gt; Frame Count</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:COUNT?
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count all received valid and invalid frames (RX).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:MPLS ON</p> <p>SENS:DATA:TEL:ETH:STR:MPLS:RX:COUN?</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ELECtrical:STReam:MPLS:LABel

---

## SCPI Command Reference

### MPLS

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:BANDwidth?

---

<b>Description</b>	<p>This query returns the frame bandwidth of Multi Protocol Label Switching (MPLS) in megabits per second.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Results &gt; Streams &gt; MPLS &gt; Total TX MPLS &gt; Ethernet BW(%)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:BANDwidth?
<b>Response Syntax</b>	<Bandwidth>
<b>Response(s)</b>	<p><b>Bandwidth:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the ethernet bandwidth (TX).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:MPLS ON</p> <p>SENS:DATA:TEL:ETH:STR:MPLS:TX:BAND?</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSExp?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:RATE?**

<b>Description</b>	<p>This query returns the Multi Protocol Label Switching (MPLS) frame rate in frame per second.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Results &gt; Streams &gt; MPLS &gt; Total TX MPLS &gt; Frame Rate (Frame/s)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:RATE?
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frame rate (TX).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:MPLS ON</p> <p>SENS:DATA:TEL:ETH:STR:MPLS:TX:RATE?</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:COSExp

---

## SCPI Command Reference

### MPLS

---

---

#### :SENSe[1..n]:DATA:TELecom:ETHernet:STReam:MPLS:TX:UTILization?

---

<b>Description</b>	<p>This query returns the Multi Protocol Label Switching (MPLS) frame utilization in percentage.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Results &gt; Streams &gt; MPLS &gt; Total TX MPLS &gt; Line Utilization(%)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:ETHernet:STReam:MPLS:TX:UTILization?
<b>Response Syntax</b>	<Utilization>
<b>Response(s)</b>	<p><b>Utilization:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frame utilization (TX).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:MPLS ON</p> <p>SENS:DATA:TEL:ETH:STR:MPLS:TX:UTIL?</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:ETHernet:STReam:MPLS:TTL?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:COUNT?**

<b>Description</b>	<p>This query returns total of all Multi Protocol Label Switching (MPLS) received valid and invalid frames.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Results &gt; Streams &gt; MPLS &gt; Total TX MPLS &gt; Frame Count</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:COUNT?
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of all received valid and invalid frames (TX).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:PORT:MPLS ON</p> <p>SENS:DATA:TEL:ETH:STR:MPLS:TX:COUN?</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TTL

---

## SCPI Command Reference

### MPLS

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:FRAMes:TX?

---

<b>Description</b>	<p>This query returns the label value (TX) for stream.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Results &gt; Streams &gt; MPLS &gt; Stream &gt;Lable1 /Label 2 TX</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:FRAMes:TX? <wsp> <Stream>, <Label Index>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Label Index:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Label Index.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of TX frames.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:STR:MPLS:FRAM:TX? 1,1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:TX:COUNt?

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:FRAMes:RX?**

<b>Description</b>	<p>This query returns the label value (RX) for stream.</p> <p>Navigation Path: Test &gt; Traffic Gen and Mon/EtherSAM &gt; Results &gt; Streams &gt; MPLS &gt; Stream &gt;Lable1 /Label 2 RX</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:FRAMes:RX? <wsp> <Stream>, <Label Index>
<b>Parameter(s)</b>	<p><b>Stream:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Traffic Gen and Mon - Selects the stream from 1 to 16.</p> <p><b>Label Index:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Label Index.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of RX frames.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:STR:MPLS:FRAM:RX? 1,1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:MPLS:RX:COUNT?

# SDT (Multi-Channel OTN)

---

**:FETCh[1..n]:DATA:TELEcom:SDT:LOTImestamp?**

---

<b>Description</b>	<p>This query returns the longest disruption detection timestamp, since test was started/reset. It is only available in the Multi-Channel OTN test.</p> <p>Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:SDT:LOTImestamp? &lt;wsp&gt; &lt;Channel&gt;</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>This parameter is mandatory in the Multi-Channel OTN application, and cannot be specified in other applications.</p> <p>The query returns the detection timestamp of the longest disruption for the specified channel. The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;Longest timestamp&gt;</p>
<b>Response(s)</b>	<p><b>Longest timestamp:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the longest disruption detection timestamp (local time), since test was started/reset.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:SDT:LOTI? 3</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:SDT:LONGest? 3</p>

---

---

**:FETCh[1..n]:DATA:TELEcom:SDT:LOCHannel?**

---

<b>Description</b>	<p>This query returns the channel with the longest disruption duration, since test was started/reset.</p> <p>It is only available in the Multi-Channel OTN test.</p> <p>Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDT:LOCHannel?
<b>Response Syntax</b>	<Channel>
<b>Response(s)</b>	<p><b>Channel:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the channel with the longest disruption duration, since test was started/reset.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDT:LOCH?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDT:LONGest? 3

---

## SCPI Command Reference

### *SDT (Multi-Channel OTN)*

---

---

#### **:FETCh[1..n]:DATA:TELeom:SDT:LATImestamp?**

<b>Description</b>	<p>This query returns the last disruption detection timestamp, since test was started/reset. It is only available in the Multi-Channel OTN test.</p> <p>Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELeom:SDT:LATImestamp? &lt;wsp&gt; &lt;Channel&gt;</code>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>This parameter is mandatory in the Multi-Channel OTN application, and cannot be specified in other applications.</p> <p>The query returns the detection timestamp of the last disruption for the specified channel.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<code>&lt;Last timestamp&gt;</code>
<b>Response(s)</b>	<p><b>Last timestamp:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the last disruption detection timestamp (local time), since test was started/reset.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:SDT:LATI? 3</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELeom:SDT:LAST? 3</code>

---

---

**:FETCh[1..n]:DATA:TELEcom:SDT:LACHannel?**

---

<b>Description</b>	This query returns the channel with the last disruption, since test was started/reset. It is only available in the Multi-Channel OTN test. Navigation Path: Test > Multi-Channel OTN >Results>SDT
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDT:LACHannel?
<b>Response Syntax</b>	<Channel>
<b>Response(s)</b>	<b>Channel:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the channel with last detected disruption, since test was started/reset.
<b>Example(s)</b>	FETC:DATA:TEL:SDT:LACH?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDT:LAST? 3

---

## SCPI Command Reference

### SDT (Multi-Channel OTN)

---

#### :FETCh[1..n]:DATA:TELecom:SDT:CHDIruption?

<b>Description</b>	<p>This query returns the number of channels that had one or more disruption(s), since test was started/reset.</p> <p>It is only available in the Multi-Channel OTN test.</p> <p>Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:SDT:CHDIruption?
<b>Response Syntax</b>	<Number of channels in disruption>
<b>Response(s)</b>	<p><b>Number of channels in disruption:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of channels that had one or more disruption(s), since test was started/reset.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDT:CHDI?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:SDT:LAST? 3 FETCh[1..n]:DATA:TEL:SDT:LACH?

---

---

**:FETCh[1..n]:DATA:TELeom:SDT:CHAThreshold?**

---

<b>Description</b>	<p>This query returns the number of channels that crossed the pass/fail verdict threshold, when SDT pass/fail verdict is enabled, since test was started/reset.</p> <p>It is only available in the Multi-Channel OTN test.</p> <p>Navigation Path: Test &gt; Multi-Channel OTN &gt;Results&gt;SDT</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:SDT:CHAThreshold?
<b>Response Syntax</b>	<Number of channels that crossed the pass/fail verdict threshold>
<b>Response(s)</b>	<p><b>Number of channels that crossed the pass/fail verdict threshold:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of channels that crossed the pass/fail verdict threshold, when SDT pass/fail verdict is enabled, since test was started/reset.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDT:CHAT?
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:SDT:VERD? 3

---

## SCPI Command Reference

*SDT (Multi-Channel OTN)*

---

**:FETCh[1..n]:DATA:TELeom:SDT:CHMOnitored?**

<b>Description</b>	This query returns the number of channels monitored. It is only available in the Multi-Channel OTN test. Navigation Path: Test > Multi-Channel OTN >Results>SDT
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:SDT:CHMOnitored?
<b>Response Syntax</b>	<Number of channels monitored>
<b>Response(s)</b>	<b>Number of channels monitored:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the number of channels monitored.
<b>Example(s)</b>	FETC:DATA:TEL:SDT:CHMO?

---



# APS

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE**

---

<b>Description</b>	<p>This command sets the switching mode</p> <p>At *RST condition, this value is set to Linear.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; TX&gt;Switching Mode</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE &lt;wsp&gt;LINEAR   RING</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LINEAR   RING</p> <p>Selects the Switching Mode.</p> <p>LINEAR: the Linear switching mode.</p> <p>RING: the Ring Switching mode.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ADV:APS:SMODE LINEAR SOUR:DATA:TEL:SDHS:ADV:APS:SMODE? Returns: LINEAR</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel</code>

---

## SCPI Command Reference

APS

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE?**

<b>Description</b>	<p>This query returns the switching mode</p> <p>At *RST condition, this value is set to Linear.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; TX&gt;Switching Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Switching mode for RS-MS advance.</p> <p>LINEAR - Linear is selected for Switching Mode.</p> <p>RING - Ring is selected for Switching Mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:SMODE LINEAR</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:SMODE?</p> <p>Returns: LINEAR</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest**

---

**Description**

This command selects the K1 Request for Linear Switching Mode.

At \*RST condition, this value is set to NREQUEST0000.

Navigation Path: Test Setup > OTN SONET/SDH BERT OR SONET/SDH BERT > Functions > APS > TX > Switching Mode (Linear) > Request

**Syntax**

:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest  
<wsp>NREQUEST0000 | DNREVERT0001 | RREQUEST0010 | UNUSED0011 |  
EXERCISER0100 | UNUSED0101 | WTRESTORE0110 | UNUSED0111 | MSWITCH1000 |  
UNUSED1001 | SDLOW1010 | SDHIGH1011 | SFLOW1100 | SFHIGH1101 |  
FSWITCH1110 | LPROTECTION1111

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest**

<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NREQUEST0000   DNREVERT0001   RREQUEST0010   UNUSED0011   EXERCISER0100   UNUSED0101   WTRESTORE0110   UNUSED0111   MSWITCH1000   UNUSED1001   SDLOW1010   SDHIGH1011   SFLOW1100   SFHIGH1101   FSWITCH1110   LPROTECTION1111</p> <p>Selects the Linear Request</p> <p>NREQUEST0000: the No Request (0000) request type.</p> <p>DNREVERT0001: the Do Not Revert (0001) request type.</p> <p>RREQUEST0010: the Reverse Request (0010) request type.</p> <p>UNUSED0011: the unused (0011) request type.</p> <p>EXERCISER0100: the Exerciser (0100) request type.</p> <p>UNUSED0101: the Unused (0101) request type.</p> <p>WTRESTORE0110: the Wait to restore (0110) request type.</p> <p>UNUSED0111: the Unused (0111) request type.</p> <p>MSWITCH1000: the Manual Switch request type.</p> <p>UNUSED1001: the Unused (1001) request type.</p> <p>SDLOW1010: the Signal Degrade (1010) request type.</p> <p>SDHIGH1011: the Signal Degrade - High Priority (1011) request type.</p> <p>SFLOW1100: the Signal Fail - Low Priority (1100) request type.</p> <p>SFHIGH1101: the Signal Fail - High Priority (1101) request type.</p> <p>FSWITCH1110: the Forced Switch (1110) request type.</p> <p>LPROTECTION1111: the Lockout of Protection (1111) request type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:LIN:REQ RREQUEST0010</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:LIN:REQ?</p> <p>Returns: RREQUEST0010</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest?**

---

<b>Description</b>	<p>This query returns the K1 request for Linear Switching Mode.</p> <p>At *RST condition, this value is set to NREQUEST0000.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; TX&gt;Switching Mode (Linear) &gt; Request</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest?
<b>Response Syntax</b>	<Type>

---

## SCPI Command Reference

APS

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest?**

<b>Response(s)</b>	<b>Type:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Linear Request NREQUEST0000 - No Request (0000) request type is selected. DNREVERT0001 - Do Not Revert (0001) request type is selected. RREQUEST0010 - Reverse Request (0010) request type is selected. UNUSED0011 - Unused (0011) request type is selected. EXERCISER0100 - Exerciser (0100) request type is selected. UNUSED0101 - Unused (0101) request type is selected. WTRESTORE0110 - Wait to restore (0110) request type is selected. UNUSED0111 - Unused (0111) request type is selected. MSWITCH1000 - Manual Switch request type is selected. UNUSED1001 - Unused (1001) request type is selected. SDLOW1010 - Signal Degrade (1010) request type is selected. SDHIGH1011 - Signal Degrade - High Priority (1011) request type is selected. SFLOW1100 - Signal Fail - Low Priority (1100) request type is selected. SFHIGH1101 - Signal Fail - High Priority (1101) request type is selected. FSWITCH1110 - Forced Switch (1110) request type is selected. LPROTECTION1111 - Lockout of Protection (1111) request type is selected.
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ADV:APS:K1:LIN:REQ RREQUEST0010 SOUR:DATA:TEL:SDHS:ADV:APS:K1:LIN:REQ? Returns: RREQUEST0010
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE?

---

**:SOURce[1..n]:DATA:TELeom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest****Description**

This command selects the K1 Request for Ring Switching Mode.

At \*RST condition, this value is set to NREQUEST0000.

Navigation Path: Test Setup > OTN SONET/SDH BERT OR SONET/SDH BERT > Functions > APS > TX > Switching Mode (Ring) > Request

**Syntax**

:SOURce[1..n]:DATA:TELeom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest  
<wsp>NREQUEST0000 | DNREVERT0001 | RREQUEST0010 | UNUSED0011 |  
EXERCISER0100 | UNUSED0101 | WTRESTORE0110 | UNUSED0111 | MSWITCH1000 |  
UNUSED1001 | SDLOW1010 | SDHIGH1011 | SFLOW1100 | SFHIGH1101 |  
FSWITCH1110 | LPROTECTION1111

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest**

<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NREQUEST0000   DNREVERT0001   RREQUEST0010   UNUSED0011   EXERCISER0100   UNUSED0101   WTRESTORE0110   UNUSED0111   MSWITCH1000   UNUSED1001   SDLOW1010   SDHIGH1011   SFLOW1100   SFHIGH1101   FSWITCH1110   LPROTECTION1111</p> <p>Selects the Ring Request</p> <p>NREQUEST0000: the No Request (0000) request type.</p> <p>DNREVERT0001: the Do Not Revert (0001) request type.</p> <p>RREQUEST0010: the Reverse Request (0010) request type.</p> <p>UNUSED0011: the unused (0011) request type.</p> <p>EXERCISER0100: the Exerciser (0100) request type.</p> <p>UNUSED0101: the Unused (0101) request type.</p> <p>WTRESTORE0110: the Wait to restore (0110) request type.</p> <p>UNUSED0111: the Unused (0111) request type.</p> <p>MSWITCH1000: the Manual Switch request type.</p> <p>UNUSED1001: the Unused (1001) request type.</p> <p>SDLOW1010: the Signal Degrade (1010) request type.</p> <p>SDHIGH1011: the Signal Degrade - High Priority (1011) request type.</p> <p>SFLOW1100: the Signal Fail - Low Priority (1100) request type.</p> <p>SFHIGH1101: the Signal Fail - High Priority (1101) request type.</p> <p>FSWITCH1110: the Forced Switch (1110) request type.</p> <p>LPROTECTION1111: the Lockout of Protection (1111) request type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:RING:REQ RREQUEST0010</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:RING:REQ?</p> <p>Returns: RREQUEST0010</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE</p>



**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest?**

<b>Description</b>	<p>This query returns the K1 request for Ring Switching Mode.</p> <p>At *RST condition, this value is set to NREQUEST0000.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; TX&gt;Switching Mode (Ring) &gt; Request</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest?
<b>Response Syntax</b>	<Type>

---

## SCPI Command Reference

APS

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:REQuest?**

<b>Response(s)</b>	<b>Type:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Linear Request NREQUEST0000 - No Request (0000) request type is selected. DNREVERT0001 - Do Not Revert (0001) request type is selected. RREQUEST0010 - Reverse Request (0010) request type is selected. UNUSED0011 - Unused (0011) request type is selected. EXERCISER0100 - Exerciser (0100) request type is selected. UNUSED0101 - Unused (0101) request type is selected. WTRESTORE0110 - Wait to restore (0110) request type is selected. UNUSED0111 - Unused (0111) request type is selected. MSWITCH1000 - Manual Switch request type is selected. UNUSED1001 - Unused (1001) request type is selected. SDLOW1010 - Signal Degrade (1010) request type is selected. SDHIGH1011 - Signal Degrade - High Priority (1011) request type is selected. SFLOW1100 - Signal Fail - Low Priority (1100) request type is selected. SFHIGH1101 - Signal Fail - High Priority (1101) request type is selected. FSWITCH1110 - Forced Switch (1110) request type is selected. LPROTECTION1111 - Lockout of Protection (1111) request type is selected.
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:ADV:APS:K1:RING:REQ RREQUEST0010 SOUR:DATA:TEL:SDHS:ADV:APS:K1:RING:REQ? Returns: RREQUEST0010
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE?

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel**

<b>Description</b>	<p>This command selects the K1 Channel for Linear Switching mode.</p> <p>At *RST condition, this value is set to NULL.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;TX &gt; Switching Mode (Linear) &gt;Channel</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel <wsp>0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15</p> <p>Sets the K1 Channel for Linear Switching.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:CHAN 1</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:CHAN?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel

## SCPI Command Reference

APS

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?**

<b>Description</b>	<p>This query returns the K1 Channel for Linear Switching Mode.</p> <p>At *RST condition, this value is set to NULL.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;TX &gt;Switching Mode (Linear) &gt;Channel</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Linear switching mode Channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:CHAN 1</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:CHAN?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel**

<b>Description</b>	<p>This command sets the Protected Channel for Linear Switching Mode.</p> <p>At *RST condition, this value is set to NULL.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;TX &gt;Switching Mode (Linear) &gt;Protected Channel</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel <wsp>0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15</p> <p>Selects the K2 Protected Channel for Linear Switching.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:PCH 1</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:PCH? Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel

## SCPI Command Reference

APS

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel?**

---

<b>Description</b>	<p>This query returns the Protected Channel for Linear Switching mode.</p> <p>At *RST condition, this value is set to NULL.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;TX &gt;Switching Mode (Linear) &gt;Protected Channel</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Linear Switching mode Protected Channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:PCH 1</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:PCH?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture**

---

<b>Description</b>	<p>This command sets the Architecture for Linear Switching mode.</p> <p>At *RST condition, this value is set to 1TO1.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;TX &gt; Switching Mode (Linear) &gt;Architecture</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture &lt;wsp&gt;1TO1   1TON</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 1TO1   1TON</p> <p>Selects the K2 Architecture for Linear Switching.</p> <p>1TO1: the 1+1 Architecture.</p> <p>1TON: the 1+N Architecture.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:ADV:APS:K2:ARCH 1TO1 SOUR:DATA:TEL:SDHS:ADV:APS:K2:ARCH? Returns: 1TO1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel</pre>

---

## SCPI Command Reference

APS

---

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture?**

<b>Description</b>	<p>This query returns the Architecture for Linear switching mode.</p> <p>At *RST condition, this value is set to 1TO1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; APS &gt; TX &gt; Switching Mode (Linear) &gt; Architecture</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Linear Switching Mode Architecture.</p> <p>1TO1 - 1+1 Architecture is selected.</p> <p>1TON - 1+N Architecture is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:ARCH 1TO1</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:ARCH?</p> <p>Returns: 1TO1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel

---



---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:OMODE**

<b>Description</b>	<p>This command sets the Operation mode for Linear Switching mode.</p> <p>At *RST condition, this value is set to RESERVED000.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; APS &gt; TX &gt; Switching Mode (Linear) &gt; Operation Mode</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:OMODE          &lt;wsp&gt;RESERVED000   RESERVED001   RESERVED010   RESERVED011   UNI100            BID101   MSRDII110   MSAIS111</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RESERVED000   RESERVED001   RESERVED010   RESERVED011   UNI100   BID101   MSRDII110   MSAIS111</p> <p>Selects the K2 Operation Mode for Linear Switching.</p> <p>RESERVED000: the Reserved (000) Operation Mode.</p> <p>RESERVED001: the Reserved (001) Operation Mode.</p> <p>RESERVED010: the Reserved (010) Operation Mode.</p> <p>RESERVED011: the Reserved (011) Operation Mode.</p> <p>UNI100: the Unidirectional (100) Operatin Mode.</p> <p>BID101: the Bidirectional (101) Operation Mode.</p> <p>MSRDII110: the MS-RDI (110) Operation Mode.</p> <p>MSAIS111: the MS-AIS (111) Operation Mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:LIN:OMODE RESERVED001</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:LIN:OMODE?</p> <p>Returns: RESERVED001</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel</p>

---

## SCPI Command Reference

APS

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:OMODE?**

<b>Description</b>	<p>This query returns the Operation mode for Linear Switching mode.</p> <p>At *RST condition, this value is set to RESERVED000.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; APS &gt; TX &gt; Switching Mode (Linear) &gt; Operation Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:OMODE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Linear Switching Operation Mode.</p> <p>RESERVED000 - RESERVED000 is selected</p> <p>RESERVED001 - RESERVED001 is selected</p> <p>RESERVED010 - RESERVED010 is selected</p> <p>RESERVED011 - RESERVED011 is selected</p> <p>UNI100 - UNI100 is selected</p> <p>MSRDI110 - MSRDI110 is selected</p> <p>MSAIS111 - MSAIS111 is selected</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:LIN:OMODE RESERVED001</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:LIN:OMODE?</p> <p>Returns: RESERVED001</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE**

<b>Description</b>	<p>This command sets the Operation mode for Ring Switching mode.</p> <p>At *RST condition, this value is set to IDLE000.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; APS &gt; TX &gt; Switching Mode (Ring) &gt; Operation Mode</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE          &lt;wsp&gt;IDLE000   BRIDGED001   BSWITCHED010   ETPROTECTION011            RESERVED100   RESERVED101   MSRDII110   MSAIS111</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: IDLE000   BRIDGED001   BSWITCHED010   ETPROTECTION011   RESERVED100   RESERVED101   MSRDII110   MSAIS111</p> <p>Selects the K2 Operation Mode for Ring Switching.</p> <p>IDLE000: the IDLE(000) Operation Mode.</p> <p>BRIDGED001: the BRIDGED (001) Operation Mode.</p> <p>BSWITCHED010: the BSWITCHED (010) Operation Mode.</p> <p>ETPROTECTION011: the ETPROTECTION (011) Operation Mode.</p> <p>RESERVED100: the RESERVED(100) Operation Mode.</p> <p>RESERVED101: the RESERVED (101) Operation Mode.</p> <p>MSRDII110: the MS-RDI (110) Operation Mode.</p> <p>MSAIS111: the MS-AIS (111) Operation Mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:RING:OMODE IDLE000</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:RING:OMODE?</p> <p>Returns: IDLE000</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel</p>

---

## SCPI Command Reference

APS

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE?**

<b>Description</b>	<p>This query returns the Operation mode for Ring Switching mode.</p> <p>At *RST condition, this value is set to IDLE000.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; APS &gt; TX &gt; Switching Mode (Ring) &gt; Operation Mode</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Ring Switching Operation Mode.</p> <p>IDLE000 - IDLE000</p> <p>BRIDGED001 - BRIDGED001 is selected</p> <p>BSWITCHED010 - BSWITCHED010 is selected</p> <p>ETPROTECTION011 - ETPROTECTION011 is selected</p> <p>RESERVED100 - RESERVED100 is selected</p> <p>RESERVED101 - RESERVED101 is selected</p> <p>MSRDI110 - MSRDI110 is selected</p> <p>MSAIS111 - MSRDI111 is selected</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:RING:OMODE IDLE000</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:RING:OMODE?</p> <p>Returns: IDLE000</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?

**:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ADVanced:APS:K[1..n]:DNODE**

<b>Description</b>	<p>This command selects Destination node Id for Ring Switching mode.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; APS &gt; TX &gt; Switching Mode (Ring) &gt; Destination Node Id</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSONet:ADVanced:APS:K[1..n]:DNODE <wsp>0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15</p> <p>Selects the K1 Destination Node Id for Ring Switching</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:DNOD 1</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:DNOD?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDHSONet:ADVanced:APS:SMODE

## SCPI Command Reference

APS

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:DNODE?**

<b>Description</b>	<p>This query returns Destination node Id for Ring Switching mode.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; TX&gt;Switching Mode (Ring) &gt; Destination Node Id</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:DNODE?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Destination Node Id for Ring Switching mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:DNOD 1</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K1:DNOD?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE?

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE**

<b>Description</b>	<p>This command selects Source Node Id for Ring Switching mode.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; APS &gt; TX &gt; Switching Mode (Ring) &gt; Source Node Id</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE <wsp>0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15</p> <p>Selects the K2 Source Node Id for Ring Switching.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:SNOD 1</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:SNOD?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel

---

## SCPI Command Reference

APS

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE?**

<b>Description</b>	<p>This query returns Source Node Id for Ring Switching mode.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; TX&gt;Switching Mode (Ring) &gt; Source Node Id</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Source Node Id for Ring Switching mode.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:SNOD 1</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:SNOD?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?



---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest**

---

<b>Description</b>	<p>This command sets Bridge Request for Ring Switching mode.</p> <p>At *RST condition, this value is set to SPATH.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;TX &gt; Switching Mode (Ring) &gt;Bridge Request</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest &lt;wsp&gt;SPATH   LPATH</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SPATH   LPATH</p> <p>Selects the K2 Bridge Request for Ring Switching.</p> <p>SPATH: Short Path Bridge Request.</p> <p>LPATH: Long Path Bridge Request.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:BRREQ SPATH</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:BRREQ?</p> <p>Returns: SPATH</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?</p>

---

## SCPI Command Reference

APS

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest?**

---

<b>Description</b>	<p>This query returns Bridge Request for Ring Switching mode.</p> <p>At *RST condition, this value is set to SPATH.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;TX &gt; Switching Mode (Ring) &gt;Bridge Request</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Bridge Request for Ring Switching mode.</p> <p>SPATH - Short Path is selected.</p> <p>LPATH - Long Path is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:BREQ SPATH</p> <p>SOUR:DATA:TEL:SDHS:ADV:APS:K2:BREQ?</p> <p>Returns: SPATH</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE**

<b>Description</b>	<p>This command selects the Switching mode for RX.</p> <p>At *RST condition, this value is set to LINEAR.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; RX&gt;Switching Mode</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:SMODE <wsp>LINEAR   RING
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LINEAR   RING</p> <p>Selects the Switching Mode.</p> <p>LINEAR: the Linear switching mode.</p> <p>RING: the Ring Switching mode.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:ADV:APS:SMODE LINEAR</p> <p>SENS:DATA:TEL:SDHS:ADV:APS:SMODE?</p> <p>Returns: LINEAR</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:OMODE

---

## SCPI Command Reference

APS

---

**:SENSe[1..n]:DATA:TELeom:SDHSonet:ADVanced:APS:SMODE?**

<b>Description</b>	<p>This query returns the Switching mode for RX.</p> <p>At *RST condition, this value is set to LINEAR.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; APS &gt; RX &gt; Switching Mode</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:SDHSonet:ADVanced:APS:SMODE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Switching mode for RS-MS advance.</p> <p>LINEAR - Linear is selected for Switching Mode.</p> <p>RING - Ring is selected for Switching Mode.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:SDHS:ADV:APS:SMODE LINEAR</p> <p>SENS:DATA:TEL:SDHS:ADV:APS:SMODE?</p> <p>Returns: LINEAR</p>
<b>See Also</b>	SENS[1..n]:DATA:TEL:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest?**

<b>Description</b>	<p>This query returns the K1 Request for Linear Switching mode for RX.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; RX&gt; Request</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:REQuest?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Request for Linear Switching mode</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ADV:APS:K1:LIN:REQ?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHS:ADV:APS:K[1..n]:BREQ?

---

## SCPI Command Reference

APS

---

---

### :FETCh[1..n]:DATA:TELEcom:SDHSONet:ADVanced:APS:K[1..n]:RING:REQuest?

---

<b>Description</b>	<p>This query returns the K1 Request for Ring Switching mode for RX.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; RX&gt; Request</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSONet:ADVanced:APS:K[1..n]:RING:REQuest?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Request for Ring Switching mode</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ADV:APS:K1:RING:REQ?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHS:ADV:APS:K[1..n]:BREQ?

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?**

<b>Description</b>	<p>This query returns Channel for Linear Switching mode RX.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;RX &gt; Switching Mode (Linear) &gt;Channel</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:CHANnel?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Channel for Linear Switching Mode.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ADV:APS:K1:CHAN?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHS:ADV:APS:K[1..n]:SNODE?

---

## SCPI Command Reference

APS

---

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel?**

---

<b>Description</b>	<p>This query returns Protected Channel for Linear Switching mode RX.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;RX &gt;Switching Mode (Linear) &gt;Protected Channel</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:PCHannel?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Protected Channel for Linear Switching mode.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ADV:APS:K2:PCH?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHS:ADV:APS:K[1..n]:DNODE?

---



**:FETCh[1..n]:DATA:TELecom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture?**

<b>Description</b>	<p>This query returns Architecture for Linear Switching mode RX.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;TX &gt; Switching Mode (Linear) &gt;Architecture</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:SDHSonet:ADVanced:APS:K[1..n]:ARCHitecture?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Architecture for Linear Switching mode.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ADV:APS:K2:ARCH?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:SDHS:ADV:APS:K[1..n]:CHAN?

---

## SCPI Command Reference

APS

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:OMODE?**

<b>Description</b>	<p>This query returns Operation mode for Linear Switching mode RX.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;RX &gt; Operation Mode</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:LINear:OMODE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Operation mode for Linear Switching mode.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ADV:APS:K2:LIN:OMODE?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHS:ADV:APS:K[1..n]:BREQ?

---

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE?**

---

<b>Description</b>	<p>This query returns Operation mode for Ring Switching mode RX.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;RX &gt; Operation Mode</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:RING:OMODE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Operation mode for Ring Switching mode.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ADV:APS:K2:RING:OMODE?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHS:ADV:APS:K[1..n]:BREQ?

---

## SCPI Command Reference

APS

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:DNODE?**

<b>Description</b>	<p>This query returns Destination node Id for Ring Switching mode RX.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; RX&gt;Switching Mode (Ring) &gt; Destination Node Id</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:DNODE?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Destination Node Id for Ring Switching mode.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ADV:APS:K1:DNODE?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHS:ADV:APS:K[1..n]:BREQ?

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE?**

<b>Description</b>	<p>This query returns Source Node Id for Ring Switching mode RX.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS&gt; RX&gt;Switching Mode (Ring) &gt; Source Node Id</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:SNODE?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR3 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Source Node Id for Ring Switching mode.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ADV:APS:K2:SNODE?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHS:ADV:APS:K[1..n]:SNODE?

---

## SCPI Command Reference

APS

---

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest?**

---

<b>Description</b>	<p>This query returns Bridge Request for Ring Switching mode RX.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; APS &gt;RX &gt; Switching Mode (Ring) &gt;Bridge Request</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:ADVanced:APS:K[1..n]:BREQuest?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Bridge Request for Ring Switching mode.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:ADV:APS:K2:BRQ?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:SDHS:ADV:APS:K[1..n]:DNODE?

---

## OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead**

---

<b>Description</b>	<p>This command sets the Optical Transport Unit (OTU) overhead byte values for the transmitter.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead &lt;wsp&gt;OA111   OA112   OA113   OA214   OA215   OA216   MFAS17   SM18   SM19   SM110   GCC0111   GCC0112   RES113   RES114, &lt;Value&gt;</p>

---

## SCPI Command Reference

OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead**

<b>Parameter(s)</b>	<b>Overhead:</b> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OA111   OA112   OA113   OA214   OA215   OA216   MFAS17   SM18   SM19   SM110   GCC0111   GCC0112   RES113   RES114</p> <p>Selects OTU overhead bytes.</p> <p>OA111</p> <p>OA112</p> <p>OA113</p> <p>OA214</p> <p>OA215</p> <p>OA216</p> <p>MFAS17</p> <p>SM18</p> <p>SM19</p> <p>SM110</p> <p>GCC0111</p> <p>GCC0112</p> <p>RES113</p> <p>RES114</p> <p>Note: The combination of row and column numbers is used along with the overhead byte. For example: OA112, here OA1 is the overhead byte, 1 is the row number, and 2 is the column number.</p> <b>Value:</b> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Optical Transport Unit (OTU) overhead byte values.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:OH:OTU1:OVER OA111, #HF6 SOUR:DATA:TEL:OTN:OH:OTU1:OVER? OA111 Returns: 246</pre>



**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead**

**See Also**

SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead?

---

## SCPI Command Reference

### OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead?**

**Description** This query returns the Optical Transport Unit (OTU) overhead byte values for the transmitter. At \*RST condition, this value is set to device-dependent.

Navigation Path: Test > Functions > OH

**Syntax** :SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead? <wsp>OA111 | OA112 | OA113 | OA214 | OA215 | OA216 | MFAS17 | SM18 | SM19 | SM110 | GCC0111 | GCC0112 | RES113 | RES114

**Parameter(s)** **Overhead:**  
The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: OA111 | OA112 | OA113 | OA214 | OA215 | OA216 | MFAS17 | SM18 | SM19 | SM110 | GCC0111 | GCC0112 | RES113 | RES114

Selects OTU overhead bytes.

OA111

OA112

OA113

OA214

OA215

OA216

MFAS17

SM18

SM19

SM110

GCC0111.

GCC0112

RES113

RES114

Note: The combination of row and column numbers is used along with the overhead byte. For example: OA112, here OA1 is the overhead byte, 1 is the row number, and 2 is the column number.

**Response Syntax** <Value>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the OTU overhead byte values in hexadecimal format. OA111, OA111 is selected as the overhead byte. OA112, OA112 is selected as the overhead byte. OA113, OA113 is selected as the overhead byte. OA214, OA214 is selected as the overhead byte. OA215, OA215 is selected as the overhead byte. OA216, OA216 is selected as the overhead byte. MFAS17, MFAS17 is selected as the overhead byte. SM18, SM18 is selected as the overhead byte. SM19, SM19 is selected as the overhead byte. SM110, SM110 is selected as the overhead byte. GCC0111, GCC0111 is selected as the overhead byte. GCC0112, GCC0112 is selected as the overhead byte. RES113, RES113 is selected as the overhead byte. RES114, RES114 is selected as the overhead byte.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OH:OTU1:OVER OA111, #HF6 SOUR:DATA:TEL:OTN:OH:OTU1:OVER? OA111 Returns: 246
<b>Note(s)</b>	<HEXADECIMAL NUMERIC RESPONSE DATA>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead?

---

## SCPI Command Reference

OH - OTN

---

**:SENSe[1..n]:DATA:TELeom:OTN:OH:OTU[1..n]:OVERhead?**

<b>Description</b>	<p>This query returns the Optical Transport Unit (OTU) overhead byte values for the receiver.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELeom:OTN:OH:OTU[1..n]:OVERhead? &lt;wsp&gt;OA111   OA112   OA113   OA214   OA215   OA216   MFAS17   SM18   SM19   SM110   GCC0111   GCC0112   RES113   RES114</p>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OA111   OA112   OA113   OA214   OA215   OA216   MFAS17   SM18   SM19   SM110   GCC0111   GCC0112   RES113   RES114</p> <p>Selects OTU overhead bytes.</p> <p>OA111</p> <p>OA112</p> <p>OA113</p> <p>OA214</p> <p>OA215</p> <p>OA216</p> <p>MFAS17</p> <p>SM18</p> <p>SM19</p> <p>SM110</p> <p>GCC0111</p> <p>GCC0112</p> <p>RES113.</p> <p>RES114</p> <p>Note: The combination of row and column numbers is used along with the overhead byte. For example: OA112, here OA1 is the overhead byte, 1 is the row number, and 2 is the column number.</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the OTU overhead byte values in hexadecimal format. OA111, OA111 is selected as the overhead byte. OA112, OA112 is selected as the overhead byte. OA113, OA113 is selected as the overhead byte. OA214, OA214 is selected as the overhead byte. OA215, OA215 is selected as the overhead byte. OA216, OA216 is selected as the overhead byte. MFAS17, MFAS17 is selected as the overhead byte. SM18, SM18 is selected as the overhead byte. SM19, SM19 is selected as the overhead byte. SM110, SM110 is selected as the overhead byte. GCC0111, GCC0111 is selected as the overhead byte. GCC0112, GCC0112 is selected as the overhead byte. RES113, RES113 is selected as the overhead byte. RES114, RES114 is selected as the overhead byte.
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OH:OTU3:OVER? OA111
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead

---

## SCPI Command Reference

### OH - OTN

---

---

#### :SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead

---

<b>Description</b>	<p>This command sets the Optical Data Unit (ODU) overhead byte values for the transmitter.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead &lt;wsp&gt;RES21   RES22   PMTCM   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPCC46   APSPCC47   APSPCC48   RES49   RES410   RES411   RES412   RES413   RES414, &lt;Value&gt;</pre>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RES21   RES22   PMTCM   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPCC46   APSPCC47   APSPCC48   RES49   RES410   RES411   RES412   RES413   RES414</p> <p>Selects ODU overhead bytes.</p> <p>Note: The combination of row and column numbers is used along with the overhead byte. For example: TCM625, here TCM6 is the overhead byte, 2 is the row number, and 5 is the column number.</p> <p><b>Value:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Optical Data Unit (ODU) overhead byte values.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:OH:ODU1:OVER RES21, #HF6 SOUR:DATA:TEL:OTN:OH:ODU1:OVER? RES21</pre> <p>Returns: 246</p>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead?</pre>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead?**

---

<b>Description</b>	<p>This query returns the Optical Data Unit (ODU) overhead byte values for the transmitter.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead? &lt;wsp&gt;RES21   RES22   PMTCM   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPC46   APSPC47   APSPC48   RES49   RES410   RES411   RES412   RES413   RES414</p>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RES21   RES22   PMTCM   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPC46   APSPC47   APSPC48   RES49   RES410   RES411   RES412   RES413   RES414</p> <p>Selects ODU overhead bytes.</p> <p>Note: The combination of row and column numbers is used along with the overhead byte. For example: TCM625, here TCM6 is the overhead byte, 2 is the row number, and 5 is the column number.</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

## SCPI Command Reference

OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the ODU overhead byte values in hexadecimal format. RES21, RES22, PMTCM, TCMACT24, TCM625, TCM626, TCM627, TCM528, TCM529, TCM5210, TCM4211, TCM4212, TCM4213, FTFL214, TCM331, TCM332, TCM333, TCM234, TCM235, TCM236, TCM137, TCM138, TCM139, PM310, PM311, PM312, EXP313, EXP314, GCC141, GCC142, GCC243, GCC244, APSPCC45, APSPCC46, APSPCC47, APSPCC48, RES49, RES410, RES411, RES412, RES413, or RES414 is selected.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OH:ODU1:OVER RES21, #HF6 SOUR:DATA:TEL:OTN:OH:ODU1:OVER? RES21 Returns: 246
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead

---



**:SENSe[1..n]:DATA:TELeom:OTN:OH:ODU[1..n]:OVERhead?**

<b>Description</b>	<p>This query returns the Optical Transport Unit (OTU) overhead byte values for the receiver.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELeom:OTN:OH:ODU[1..n]:OVERhead? &lt;wsp&gt;RES21   RES22   PMTCM   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPCC46   APSPCC47   APSPCC48   RES49   RES410   RES411   RES412   RES413   RES414</pre>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RES21   RES22   PMTCM   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPCC46   APSPCC47   APSPCC48   RES49   RES410   RES411   RES412   RES413   RES414</p> <p>Selects ODU overhead bytes.</p> <p>RES21, RES22, PMTCM, TCMACT24, TCM625, TCM626, TCM627, TCM528, TCM529, TCM5210, TCM4211, TCM4212, TCM4213, FTFL214, TCM331, TCM332, TCM333, TCM234, TCM235, TCM236, TCM137, TCM138, TCM139, PM310, PM311, PM312, EXP313, EXP314, GCC141, GCC142, GCC243, GCC244, APSPCC45, APSPCC46, APSPCC47, APSPCC48, RES49, RES410, RES411, RES412, RES413, or RES414.</p> <p>Note: The combination of row and column numbers is used along with the overhead byte. For example: TCM625, here TCM6 is the overhead byte, 2 is the row number, and 5 is the column number.</p>
<b>Response Syntax</b>	<Value>

## SCPI Command Reference

*OH - OTN*

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the ODU overhead byte values in hexadecimal format. RES21, PMTCM, TCMACT24, TCM625, TCM626, TCM627, TCM528, TCM529, TCM5210, TCM4211, TCM4212, TCM4213, FTFL214, TCM331, TCM332, TCM333, TCM234, TCM235, TCM236, TCM137, TCM138, TCM139, PM310, PM311, PM312, EXP313, EXP314, GCC141, GCC142, GCC243, GCC244, APSPCC45, APSPCC46, APSPCC47, APSPCC48, RES49, RES410, RES411, RES412, RES413, or RES414.
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OH:ODU1:OVER? RES21
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead**

<b>Description</b>	<p>This command sets the Optical Payload Unit (OPU) overhead byte values for the transmitter.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead <wsp>JC6   JC5   JC4   JC3   JC2   JC1   RES416   JC116   NJO416   OMFI, <Value>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: JC6   JC5   JC4   JC3   JC2   JC1   RES416   JC116   NJO416   OMFI</p> <p>Selects OPU overhead bytes.</p> <p>JC6</p> <p>JC5</p> <p>JC4</p> <p>JC3</p> <p>JC2</p> <p>JC1</p> <p>RES416</p> <p>JC116</p> <p>NJO416</p> <p>OMFI</p> <p><b>Value:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Optical Payload Unit (OPU) overhead byte values.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OH:OPU3:OVER JC5, #HF6</p> <p>SOUR:DATA:TEL:OTN:OH:OPU3:OVER? JC5</p> <p>Returns: 246</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead?

## SCPI Command Reference

OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead?**

<b>Description</b>	<p>This command sets the Optical Payload Unit (OPU) overhead byte values for the transmitter.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead? &lt;wsp&gt;JC6   JC5   JC4   JC3   JC2   JC1   RES416   JC116   NJO416   OMFI</p>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: JC6   JC5   JC4   JC3   JC2   JC1   RES416   JC116   NJO416   OMFI</p> <p>Selects OPU overhead bytes.</p> <p>JC6</p> <p>JC5</p> <p>JC4</p> <p>JC3</p> <p>JC2</p> <p>JC1</p> <p>RES416</p> <p>JC116</p> <p>NJO416</p> <p>OMFI</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the OPU overhead byte values in hexadecimal format. JC6, JC6 is selected as the overhead byte. JC5, JC5 is selected as the overhead byte. JC4, JC4 is selected as the overhead byte. JC3, JC3 is selected as the overhead byte. JC2, JC2 is selected as the overhead byte. JC1, JC1 is selected as the overhead byte. RES416, RES416 is selected as the overhead byte. JC116, JC116 is selected as the overhead byte. NJO416, NJO416 is selected as the overhead byte. OMFI, OMFI is selected as the overhead byte.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OH:OPU3:OVER JC5, #HF6 SOUR:DATA:TEL:OTN:OH:OPU3:OVER? JC5 Returns: 246
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead

---

## SCPI Command Reference

*OH - OTN*

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead?**

<b>Description</b>	This query returns the Optical Payload Unit (OPU) overhead byte values for the transmitter. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Functions > OH
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead? <wsp>RES115   RES215   RES315   RES116   JC116   RES216   JC216   RES316   JC316   NJO416   JC6   JC5   JC4   JC3   JC2   JC1   RES416   JC116   OMFI

---

:SENSe[1..n]:DATA:TELeCom:OTN:OH:OPU[1..n]:OVERhead?

**Parameter(s)****Overhead:**

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: RES115 | RES215 | RES315 | RES116 | JC116 | RES216 | JC216 | RES316 | JC316 | NJO416 | JC6 | JC5 | JC4 | JC3 | JC2 | JC1 | RES416 | JC116 | OMFI

Selects OPU overhead bytes.

RES115

RES215

RES315

RES116

JC116

RES216

JC216

RES316

JC316

NJO416

JC6

JC5

JC4

JC3

JC2

JC1

RES416

JC116

OMFI

Note: The combination of row and column number is used along with the overhead byte. For Example: RES115, here RES is the overhead byte, 1 is the row number, and 15 is the column number.

**Response Syntax**

&lt;Value&gt;

## SCPI Command Reference

OH - OTN

---

**:SENSe[1..n]:DATA:TELecom:OTN:OH:OPU[1..n]:OVERhead?**

<b>Response(s)</b>	<b>Value:</b>
	<p>The response data syntax for this parameter is defined as a &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the OPU overhead byte values in hexadecimal format.</p> <p>RES115, RES115 is selected as the overhead byte.</p> <p>RES215, RES215 is selected as the overhead byte.</p> <p>RES315, RES315 is selected as the overhead byte.</p> <p>RES116, RES116 is selected as the overhead byte.</p> <p>JC116, JC116 as is selected the overhead byte.</p> <p>RES216, RES216 is selected as the overhead byte.</p> <p>JC216, JC216 is selected as the overhead byte.</p> <p>RES316, RES316 is selected as the overhead byte.</p> <p>JC316, JC316 is selected as the overhead byte.</p> <p>NJO416, NJO416 is selected as overhead byte.</p> <p>JC6, JC6 is selected as the overhead byte.</p> <p>JC5, JC5 is selected as the overhead byte.</p> <p>JC4, JC4 is selected as the overhead byte.</p> <p>JC3, JC3 is selected as the overhead byte.</p> <p>JC2, JC2 is selected as the overhead byte.</p> <p>JC1, JC1 is selected as the overhead byte.</p> <p>RES416, RES416 is selected as the overhead byte.</p> <p>JC116, JC16 is selected as the overhead byte.</p> <p>OMFI, OMFI is selected as the overhead byte.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OH:OPU:OVER? JC5
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:OTN:OH:ODU[1..n]:OVERhead



**:SOURce[1..n]:DATA:TELEcom:OTN:OH:REStore:DEFault**

---

<b>Description</b>	<p>This command resets or overwrites the OTN overhead byte values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; OH&gt; Default OTN OH</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OH:REStore:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OH:REStore:DEFault
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead?

---

## SCPI Command Reference

*OH - OTN*

---

**:SOURce[1..n]:DATA:TELeom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead**

<b>Description</b>	This command sets the overhead byte values for overclocked rates OTU3e1/2. At *RST condition, this value is device dependent. Navigation Path: Test > Functions > OH
<b>Syntax</b>	:SOURce[1..n]:DATA:TELeom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead <wsp>OA111   OA112   OA113   OA214   OA215   OA216   MFAS17   SM18   SM19   SM110   GCC0111   GCC0112   RES113   RES114, <Value>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead**

<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OA111   OA112   OA113   OA214   OA215   OA216   MFAS17   SM18   SM19   SM110   GCC0111   GCC0112   RES113   RES114</p> <p>Selects overhead bytes for non standard rates OTU3e1/2.</p> <p>OA111</p> <p>OA112</p> <p>OA113</p> <p>OA214</p> <p>OA215</p> <p>OA216</p> <p>MFAS17</p> <p>SM18</p> <p>SM19</p> <p>SM110</p> <p>GCC0111</p> <p>GCC0112</p> <p>RES113</p> <p>RES114</p> <p>Note: The combination of row and column number is used along with overhead byte. For Ex: OA112, here OA1 is overhead byte, 1 is row number, and 2 is column number.</p> <p><b>Value:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the overhead byte values.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OH:OTU3:E1:OVER OA111, #HF6</p> <p>SOUR:DATA:TEL:OTN:OH:OTU3:E1:OVER? OA111</p> <p>Returns: 246</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead?</p>

## SCPI Command Reference

### OH - OTN

---

**:SOURce[1..n]:DATA:TELecom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead?**

<b>Description</b>	<p>This query returns the overhead byte values for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELecom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead? &lt;wsp&gt;OA111   OA112   OA113   OA214   OA215   OA216   MFAS17   SM18   SM19   SM110   GCC0111   GCC0112   RES113   RES114</p>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OA111   OA112   OA113   OA214   OA215   OA216   MFAS17   SM18   SM19   SM110   GCC0111   GCC0112   RES113   RES114</p> <p>Selects overhead bytes for non standard rates OTU3e1/2.</p> <p>OA111 OA112 OA113 OA214 OA215 OA216 MFAS17 SM18 SM19 SM110 GCC0111 GCC0112 RES113 RES114</p> <p>Note: The combination of row and column number is used along with overhead byte. For Ex: OA112, here OA1 is overhead byte, 1 is row number, and 2 is column number.</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead?**

---

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the overhead byte values in hexadecimal format.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OH:OTU3:E1:OVER OA111, #HF6 SOUR:DATA:TEL:OTN:OH:OTU3:E1:OVER? OA111 Returns: 246
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead

---

## SCPI Command Reference

### OH - OTN

---

**:SENSe[1..n]:DATA:TELeom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead?**

<b>Description</b>	<p>This query returns the overhead byte values for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELeom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead? &lt;wsp&gt;OA111   OA112   OA113   OA214   OA215   OA216   MFAS17   SM18   SM19   SM110   GCC0111   GCC0112   RES113   RES114</p>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OA111   OA112   OA113   OA214   OA215   OA216   MFAS17   SM18   SM19   SM110   GCC0111   GCC0112   RES113   RES114</p> <p>Selects overhead bytes for non standard rates OTU3e1/2.</p> <p>OA111 OA112 OA113 OA214 OA215 OA216 MFAS17 SM18 SM19 SM110 GCC0111 GCC0112 RES113 RES114</p> <p>Note: The combination of row and column number is used along with overhead byte. For Ex: OA112, here OA1 is overhead byte, 1 is row number, and 2 is column number.</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead?**

---

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the overhead byte values in hexadecimal format.
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OH:OTU3:E1:OVER? OA111
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:E[1..n]:OVERhead

---

## SCPI Command Reference

### OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead**

<b>Description</b>	<p>This command sets the overhead byte values for overlocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead &lt;wsp&gt;RES21   RES22   RES23   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPCC46   APSPCC47   APSPCC48   RES49   RES410   RES411   RES412   RES413   RES414, &lt;Value&gt;</p>

---



**:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead**

<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RES21   RES22   RES23   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPCC46   APSPCC47   APSPCC48   RES49   RES410   RES411   RES412   RES413   RES414</p> <p>Selects overhead bytes for non standard rates OTU1e/2e.</p> <p>RES21, RES22, RES23, TCMACT24, TCM625, TCM626, TCM627, TCM528, TCM529, TCM5210, TCM4211, TCM4212, TCM4213, FTFL214, TCMACT331, TCM332, TCM333, TCM234, TCM235, TCM236, TCM137, TCM138, TCM139, PM310, PM311, PM312, EXP313, EXP314, GCC141, GCC142, GCC243, GCC244, APSPCC45, APSPCC46, APSPCC47, APSPCC48, RES49, RES410, RES411, RES412, RES413, or RES414.</p> <p>Note: The combination of row and column number is used along with overhead byte. For Ex: TCM625, here TCM6 is overhead byte, 2 is row number, and 5 is column number.</p> <p><b>Value:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the overhead byte values.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OH:ODU3:E1:OVER RES21, #HF6</p> <p>SOUR:DATA:TEL:OTN:OH:ODU3:E1:OVER? RES21</p> <p>Returns: 246</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead?</p>

## SCPI Command Reference

### OH - OTN

---

**:SOURce[1..n]:DATA:TELecom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead?**

<b>Description</b>	<p>This query returns the overhead byte values for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELecom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead? &lt;wsp&gt;RES21   RES22   RES23   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPCC46   APSPCC47   APSPCC48   RES49   RES410   RES411   RES412   RES413   RES414</p>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RES21   RES22   RES23   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPCC46   APSPCC47   APSPCC48   RES49   RES410   RES411   RES412   RES413   RES414</p> <p>Selects overhead bytes for non standard rates OTU1e/2e.</p> <p>RES21, RES22, RES23, TCMACT24, TCM625, TCM626, TCM627, TCM528, TCM529, TCM5210, TCM4211, TCM4212, TCM4213, FTFL214, TCMACT331, TCM332, TCM333, TCM234, TCM235, TCM236, TCM137, TCM138, TCM139, PM310, PM311, PM312, EXP313, EXP314, GCC141, GCC142, GCC243, GCC244, APSPCC45, APSPCC46, APSPCC47, APSPCC48, RES49, RES410, RES411, RES412, RES413, or RES414.</p> <p>Note: The combination of row and column number is used along with overhead byte. For Ex: TCM625, here TCM6 is overhead byte, 2 is row number, and 5 is column number.</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead?**

---

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the overhead byte values in hexadecimal format.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OH:ODU3:E1:OVER RES21, #HF6 SOUR:DATA:TEL:OTN:OH:ODU3:E1:OVER? RES21 Returns: 246
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead

---

## SCPI Command Reference

### OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:DEFault**

<b>Description</b>	<p>This command resets or overwrites the overhead byte values for overclocked rates OTU3e1/2.</p> <p>This command is an event and has no associated *RST condition or query form.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OH:ODU3:E1:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead?

---

**:SENSe[1..n]:DATA:TELecom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead?**

<b>Description</b>	<p>This query returns the overhead byte values for overclocked rates OTU3e1/2.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELecom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead? &lt;wsp&gt;RES21   RES22   RES23   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   GCC2   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPCC46   APSPCC47   APSPCC48   RES49   RES410   RES411   RES412   RES413   RES414</pre>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: RES21   RES22   RES23   TCMACT24   TCM625   TCM626   TCM627   TCM528   TCM529   TCM5210   TCM4211   TCM4212   TCM4213   FTFL214   GCC2   TCM331   TCM332   TCM333   TCM234   TCM235   TCM236   TCM137   TCM138   TCM139   PM310   PM311   PM312   EXP313   EXP314   GCC141   GCC142   GCC243   GCC244   APSPCC45   APSPCC46   APSPCC47   APSPCC48   RES49   RES410   RES411   RES412   RES413   RES414</p> <p>Selects overhead bytes.</p> <p>RES21, RES22, RES23, TCMACT24, TCM625, TCM626, TCM627, TCM528, TCM529, TCM5210, TCM4211, TCM4212, TCM4213, FTFL214, TCMACT331, TCM332, TCM333, TCM234, TCM235, TCM236, TCM137, TCM138, TCM139, PM310, PM311, PM312, EXP313, EXP314, GCC141, GCC142, GCC243, GCC244, APSPCC45, APSPCC46, APSPCC47, APSPCC48, RES49, RES410, RES411, RES412, RES413, or RES414.</p> <p>Note: The combination of row and column number is used along with overhead byte. For Ex: TCM625, here TCM6 is overhead byte, 2 is row number, and 5 is column number.</p>
<b>Response Syntax</b>	<Value>

## SCPI Command Reference

*OH - OTN*

---

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead?**

---

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the overhead byte values in hexadecimal format.
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OH:ODU3:E1:OVER? RES21
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:E[1..n]:OVERhead

---

---

**:SOURce[1..n]:DATA:TELecom:OTN:OH:OPU[1..n]:PSI**

---

<b>Description</b>	<p>This command sets the Optical Payload Unit (OPU) overhead byte values for the transmitter. At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH &gt; OTU/ODU &gt; TX &gt; PSI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELecom:OTN:OH:OPU[1..n]:PSI <wsp> <PSI>, <Value>
<b>Parameter(s)</b>	<p><b>PSI:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects OPU PSI Bytes.</p> <p><b>Value:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Optical Payload Unit (OPU) PSI byte values.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:OH:OPU1:PSI 115,#HF6 SOUR:DATA:TEL:OTN:OH:OPU1:PSI? 115 Returns: 246</pre>
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:OTN:OH:OPU[1..n]:PSI?

---

## SCPI Command Reference

OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI?**

<b>Description</b>	<p>This query returns the Optical Payload Unit (OPU) overhead byte values for the transmitter. At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH &gt; OTU/ODU &gt; TX &gt; PSI</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI? &lt;wsp&gt; &lt;PSI&gt;</code>
<b>Parameter(s)</b>	<p><b>PSI:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects OPU PSI Bytes.</p>
<b>Response Syntax</b>	<code>&lt;Value&gt;</code>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the OPU PSI byte values in hexadecimal format.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OH:OPU1:PSI 115,#HF6</p> <p>SOUR:DATA:TEL:OTN:OH:OPU1:PSI? 115</p> <p>Returns: 246</p>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI?</code>

---



**:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI?**

<b>Description</b>	<p>This query returns the Optical Payload Unit (OPU) overhead byte values for the receiver.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH &gt; OTU/ODU &gt; RX &gt; PSI</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI? <wsp> <PSI>
<b>Parameter(s)</b>	<p><b>PSI:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects OPU PSI Bytes.</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the OPU PSI byte values in hexadecimal format.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OH:OPU:PSI? 21
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI

---

## SCPI Command Reference

OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead?**

<b>Description</b>	<p>This command sets the Optical Payload Unit (OPU) E overhead byte values for the transmitter.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead? &lt;wsp&gt;JC6   JC5   JC4   JC3   JC2   JC1   RES416   JC116   NJO416   OMFI</p>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: JC6   JC5   JC4   JC3   JC2   JC1   RES416   JC116   NJO416   OMFI</p> <p>Selects OPU overhead bytes.</p> <p>JC6</p> <p>JC5</p> <p>JC4</p> <p>JC3</p> <p>JC2</p> <p>JC1</p> <p>RES416</p> <p>JC116</p> <p>NJO416</p> <p>OMFI</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the OPU overhead byte values in hexadecimal format. JC6, selects JC6 as the overhead byte. JC5, selects JC5 as the overhead byte. JC4, selects JC4 as the overhead byte. JC3, selects JC3 as the overhead byte. JC2, selects JC2 as the overhead byte. JC1, selects JC1 as the overhead byte. RES416, selects RES416 as the overhead byte. JC116, selects JC116 as the overhead byte. NJO416, selects NJO416 as the overhead byte. OMFI, selects OMFI as the overhead byte.
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OH:OPU1:E:OVER JC6, #HF6 SOUR:DATA:TEL:OTN:OH:OPU1:E:OVER? JC6 Returns: 246
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:OVERhead

---

## SCPI Command Reference

### OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:DEFault**

<b>Description</b>	<p>This command resets or overwrites the Optical Payload Unit (OPU) overhead byte values. This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p> <p>Note: This command is obsolete now, this will get removed in the next release.</p> <p>Correct command to use is ""SOUR:DATA:TEL:OTN:OH:REStore:DEFault"".</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OH:OPU1:E:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]:DEFault

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead?**

<b>Description</b>	<p>This query returns the Optical Payload Unit (OPU) E overhead byte values for the transmitter.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:OVERhead? &lt;wsp&gt;RES115   RES215   RES315   RES116   JC116   RES216   JC216   RES316   JC316   NJO416   JC6   JC5   JC4   JC3   JC2   JC1   RES416   JC116   OMFI</pre>

---

## SCPI Command Reference

*OH - OTN*

---

**:SENSe[1..n]:DATA:TELecom:OTN:OH:OPU[1..n]:E:OVERhead?**

### Parameter(s)

#### Overhead:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: RES115 | RES215 | RES315 | RES116 | JC116 | RES216 | JC216 | RES316 | JC316 | NJO416 | JC6 | JC5 | JC4 | JC3 | JC2 | JC1 | RES416 | JC116 | OMFI

Selects OPU overhead bytes.

RES115

RES215

RES315

RES116

JC116

RES216

JC216

RES316

JC316

NJO416

JC6

JC5

JC4

JC3

JC2

JC1

RES416

JC116

OMFI

Note: The combination of row and column number is used along with the overhead byte. For Example: RES115, here RES is the overhead byte, 1 is the row number, and 15 is the column number.

**Response Syntax** <Value>

---

---

**:SENSe[1..n]:DATA:TELecom:OTN:OH:OPU[1..n]:E:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the OPU overhead byte values in hexadecimal format. RES115, RES115 is selected as the overhead byte. RES215, RES215 is selected as the overhead byte. RES315, RES315 is selected as the overhead byte. RES116, RES116 is selected as the overhead byte. JC116, JC116 as is selected the overhead byte. RES216, RES216 is selected as the overhead byte. JC216, JC216 is selected as the overhead byte. RES316, RES316 is selected as the overhead byte. JC316, JC316 is selected as the overhead byte. NJO416, NJO416 is selected as overhead byte. JC6, JC6 is selected as the overhead byte. JC5, JC5 is selected as the overhead byte. JC4, JC4 is selected as the overhead byte. JC3, JC3 is selected as the overhead byte. JC2, JC2 is selected as the overhead byte. JC1, JC1 is selected as the overhead byte. RES416, RES416 is selected as the overhead byte. JC116, JC16 is selected as the overhead byte. OMFI, OMFI is selected as the overhead byte.
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OH:OPU1:E:OVER? RES21
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:OTN:OH:ODU[1..n]:OVERhead

---

## SCPI Command Reference

OH - OTN

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI**

<b>Description</b>	<p>This command sets the Optical Payload Unit (OPU) E overhead byte values for the transmitter.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH &gt; OTU/ODU &gt; TX &gt; PSI</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI &lt;wsp&gt; &lt;PSI&gt;, &lt;Value&gt;</code>
<b>Parameter(s)</b>	<p><b>PSI:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects OPU PSI Bytes.</p> <p><b>Value:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Optical Payload Unit (OPU) PSI byte values.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:OTN:OH:OPU1:E:PSI 115,#HF6 SOUR:DATA:TEL:OTN:OH:OPU1:E:PSI? 115 Returns: 246</pre>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI?</code>



---

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI?**

---

<b>Description</b>	<p>This query returns the Optical Payload Unit (OPU) E overhead byte values for the transmitter.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; OH &gt; OTU/ODU &gt; TX &gt; PSI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI? <wsp> <PSI>
<b>Parameter(s)</b>	<p><b>PSI:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects OPU PSI Bytes.</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the OPU PSI byte values in hexadecimal format.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OH:OPU1:E:PSI 115,#HF6</p> <p>SOUR:DATA:TEL:OTN:OH:OPU1:E:PSI? 115</p> <p>Returns: 246</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI?

---

## SCPI Command Reference

OH - OTN

---

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI?**

<b>Description</b>	This query returns the Optical Payload Unit (OPU) E overhead byte values for the receiver. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Functions > OH > OTU/ODU > RX > PSI
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:E:PSI? <wsp> <PSI>
<b>Parameter(s)</b>	<b>PSI:</b> The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects OPU PSI Bytes.
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the OPU PSI byte values in hexadecimal format.
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OH:OPU1:E:PSI? 21
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:PSI

---

## OH - SONET/SDH

---

**:SOURCE[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead**

---

<b>Description</b>	<p>This command sets the section overhead byte values in hexadecimal format.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH &gt;SONET&gt;SECTION &gt; TX</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead &lt;wsp&gt; &lt;Timeslot&gt;, A1   A2   J0   B1   E1   F1   D1   D2   D3   Z0, &lt;Value&gt;</p>

---

## SCPI Command Reference

*OH - SONET/SDH*

---

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead**

<b>Parameter(s)</b>	<p><b>Timeslot:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the section overhead timeslot number for transmitter.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: A1   A2   J0   B1   E1   F1   D1   D2   D3   Z0</p> <p>Selects the section overhead bytes.</p> <p>A1: F6 hexadecimal value</p> <p>A2: 28 hexadecimal value</p> <p>J0: J0 trace</p> <p>B1: Bit Interleaved Parity code (BIP-8)</p> <p>E1: Orderwire</p> <p>F1: User</p> <p>D1: Data Communications Channel (DCC)</p> <p>D2: Data Communications Channel (DCC)</p> <p>D3: Data Communications Channel (DCC)</p> <p>Z0: Growth Byte</p> <p><b>Value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the section overhead byte values in hexadecimal format.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SON:OH:SECT:OVER 1, A1, #HF6</p> <p>SOUR:DATA:TEL:SON:OH:SECT:OVER? 1, A1</p> <p>Returns: 246</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:LINE:OVERhead?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead?**

<b>Description</b>	<p>This query returns the section overhead byte values in decimal format.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; OH &gt; SONET &gt; SECTION &gt; TX</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead? <wsp> <Timeslot>, A1   A2   J0   B1   E1   F1   D1   D2   D3   Z0
<b>Parameter(s)</b>	<p><b>Timeslot:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the section overhead timeslot number.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: A1   A2   J0   B1   E1   F1   D1   D2   D3   Z0</p> <p>Selects the section overhead bytes.</p> <p>A1: F6 hexadecimal value  A2: 28 hexadecimal value  J0: J0 trace  B1: Bit Interleaved Parity code (BIP-8)  E1: Orderwire  F1: User  D1: Data Communications Channel (DCC)  D2: Data Communications Channel (DCC)  D3: Data Communications Channel (DCC)  Z0: Growth Byte</p>
<b>Response Syntax</b>	<Value>

## SCPI Command Reference

*OH - SONET/SDH*

---

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the section overhead byte value.
<b>Example(s)</b>	SOUR:DATA:TEL:SON:OH:SECT:OVER 1, A1, #HF6 SOUR:DATA:TEL:SON:OH:SECT:OVER? 1, A1 Returns: 246
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:LINE:OVERhead

---

**:SOURCE[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead:DEFault**

<b>Description</b>	<p>This command resets the section overhead byte values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH &gt;SONET&gt; TX &gt; SECTION&gt;Default</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead:DEFault <wsp> <Timeslot>, A1   A2   J0   B1   E1   F1   D1   D2   D3   Z0
<b>Parameter(s)</b>	<p><b>Timeslot:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the section overhead timeslot number.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: A1   A2   J0   B1   E1   F1   D1   D2   D3   Z0</p> <p>Selects the section overhead bytes.</p> <p>A1: F6 hexadecimal value</p> <p>A2: 28 hexadecimal value</p> <p>J0: J0 trace</p> <p>B1: Bit Interleaved Parity code (BIP-8)</p> <p>E1: Orderwire</p> <p>F1: User</p> <p>D1: Data Communications Channel (DCC)</p> <p>D2: Data Communications Channel (DCC)</p> <p>D3: Data Communications Channel (DCC)</p> <p>Z0: Growth Byte</p>
<b>Example(s)</b>	SOUR:DATA:TEL:SON:OH:SECT:OVER:DEF 1, A1
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDHsonet:OH:REStore:DEF

## SCPI Command Reference

### OH - SONET/SDH

---

#### :SENSe[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead?

<b>Description</b>	<p>This query returns the section overhead byte values in decimal format for the receiver.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH &gt;SONET&gt;SECTION &gt; RX</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead? <wsp> <Timeslot>, A1   A2   J0   B1   E1   F1   D1   D2   D3   Z0
<b>Parameter(s)</b>	<p><b>Timeslot:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the section overhead timeslot number for the receiver.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: A1   A2   J0   B1   E1   F1   D1   D2   D3   Z0</p> <p>Selects the section overhead bytes for the receiver.</p> <p>A1: F6 hexadecimal value</p> <p>A2: 28 hexadecimal value</p> <p>J0: J0 trace</p> <p>B1: Bit Interleaved Parity code (BIP-8)</p> <p>E1: Orderwire</p> <p>F1: User</p> <p>D1: Data Communications Channel (DCC)</p> <p>D2: Data Communications Channel (DCC)</p> <p>Z0: Growth Byte</p>
<b>Response Syntax</b>	<Value>

---



**:SENSe[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the section overhead byte value for the receiver.
<b>Example(s)</b>	SENS:DATA:TEL:SON:OH:SECT:OVER? 1, A1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDHSonet:OH:LINE:OVERhead?

---

## SCPI Command Reference

### *OH - SONET/SDH*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead**

---

<b>Description</b>	<p>This command sets the line overhead byte values in hexadecimal format.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH &gt;SONET&gt;LINE &gt; TX</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead &lt;wsp&gt; &lt;Timeslot&gt;, H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   Z1   Z2   M1   M0   S1, &lt;Value&gt;</p>

---

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead**

<b>Parameter(s)</b>	<p><b>Timeslot:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the line overhead timeslot number for transmitter.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   Z1   Z2   M1   M0   S1</p> <p>Selects the line overhead bytes.</p> <p>H1 to H3: respectively H1, H2, or H3 Pointer</p> <p>B2: Bit Interleaved Parity code (BIP-8)</p> <p>K1 and K2: respectively K1 or K2 Automatic Protection Switching (APS)</p> <p>D4 to D12: respectively D4 to D12 Data Communications Channel (DCC)</p> <p>Z1 and Z2: respectively Z1 or Z2 Growth byte</p> <p>M1: M1 Remote Error Indicator - Line (REI-L)</p> <p>M0: M0 Remote Error Indicator - Line (REI-L)</p> <p>S1: S1 byte</p> <p><b>Value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the line overhead byte values in hexadecimal format.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SON:OH:LINE:OVER 1, D4,#H00</p> <p>SOUR:DATA:TEL:SON:OH:LINE:OVER? 1, D4</p> <p>Returns: 0</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSONet:OH:SECTion:OVERhead?</p>

## SCPI Command Reference

### OH - SONET/SDH

---

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead?**

<b>Description</b>	<p>This query returns the line overhead byte values in decimal format.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH &gt;SONET&gt;LINE &gt; TX</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead? &lt;wsp&gt; &lt;Timeslot&gt;, H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   Z1   Z2   M1   M0   S1</p>
<b>Parameter(s)</b>	<p><b>Timeslot:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the line overhead timeslot number.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   Z1   Z2   M1   M0   S1</p> <p>Selects the line overhead bytes.</p> <p>H1 to H3: Respectively H1, H2, or H3 Pointer</p> <p>B2: Bit Interleaved Parity code (BIP-8)</p> <p>K1 and K2: Respectively K1 or K2 Automatic Protection Switching (APS)</p> <p>D4 to D12: Respectively D4 to D12 Data Communications Channel (DCC)</p> <p>Z1 and Z2: Respectively Z1 or Z2 Growth byte</p> <p>M1: Remote Error Indicator - Line (REI-L)</p> <p>M0: Remote Error Indicator - Line (REI-L)</p> <p>S1: S1 byte</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the line overhead byte value.
<b>Example(s)</b>	SOUR:DATA:TEL:SON:OH:LINE:OVER 1, D4,#H00 SOUR:DATA:TEL:SON:OH:LINE:OVER? 1, D4 Returns: 0
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:SECTion:OVERhead

---

## SCPI Command Reference

### OH - SONET/SDH

---

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead:DEFault**

<b>Description</b>	<p>This command resets the line overhead byte values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; OH &gt; SONET &gt; TX &gt; LINE &gt; Default</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead:DEFault &lt;wsp&gt; &lt;Timeslot&gt;, H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   Z1   Z2   M0   M1   S1</p>
<b>Parameter(s)</b>	<p><b>Timeslot:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the line overhead timeslot number.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   Z1   Z2   M0   M1   S1</p> <p>Selects the line overhead bytes.</p> <p>Selects the line overhead bytes.</p> <p>H1 to H3: Respectively H1, H2, or H3 Pointer</p> <p>B2: Bit Interleaved Parity code (BIP-8)</p> <p>K1 and K2: Respectively K1 or K2 Automatic Protection Switching (APS)</p> <p>D4 to D12: Respectively D4 to D12 Data Communications Channel (DCC)</p> <p>Z1 and Z2: Respectively Z1 or Z2 Growth byte</p> <p>M1: Remote Error Indicator - Line (REI-L)</p> <p>M0: Remote Error Indicator - Line (REI-L)</p> <p>S1: S1 byte</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SON:OH:LINE:OVER:DEF 1, H1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHsonet:OH:SECTion:OVERhead:DEFault</p>

**:SENSe[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead?**

<b>Description</b>	<p>This query returns the line overhead byte values in decimal format for the receiver.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; OH &gt; SONET &gt; LINE &gt; RX</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead? <wsp> <Timeslot>, H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   Z1   Z2   M0   M1   S1   E2
<b>Parameter(s)</b>	<p><b>Timeslot:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the line overhead timeslot number for the receiver.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   Z1   Z2   M0   M1   S1   E2</p> <p>Selects the line overhead bytes for the receiver.</p> <p>H1 to H3: Respectively H1, H2, or H3 Pointer</p> <p>B2: Bit Interleaved Parity code (BIP-8)</p> <p>K1 and K2: Respectively K1 or K2 Automatic Protection Switching (APS)</p> <p>D4 to D12: Respectively D4 to D12 Data Communications Channel (DCC)</p> <p>S1: Synchronization status</p> <p>E2: Orderwire</p> <p>Z2: Growth</p> <p>Byte is specified in two ways.</p> <p>In first method standard names are used. Ex:A1, A2.</p> <p>In second method an ""UDrc"" notation is used, where ""r"" is the numerical value of the bytes row in the transport overhead and ""c"" is the numerical value of the bytes column in the transport overhead. Ex:UD11, UD12.</p> <p>M1: Remote Error Indicator - Line (REI-L)</p> <p>M0: Remote Error Indicator - Line (REI-L)</p>

## SCPI Command Reference

### *OH - SONET/SDH*

---

**:SENSe[1..n]:DATA:TELeCom:SONet:OH:LINE:OVERhead?**

**Response Syntax** <Value>

**Response(s)** **Value:**

The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the line overhead byte value for the receiver.

**Example(s)** SENS:DATA:TEL:SON:OH:LINE:OVER? 1, D4

**See Also** SENSe[1..n]:DATA:TELeCom:SDHSONet:OH:SECTion:OVERhead?

---



---

**:SOURCE[1..n]:DATA:TELECOM:SDH:OH:RS:OVERHEAD[1..n]**

---

**Description**

This command sets the regenerator overhead byte values in hexadecimal format.

At \*RST condition, this value is device dependent.

Navigation Path: Test Setup > OTN SONET/SDH BERT OR SONET/SDH BERT >

Functions > OH > SDH > TX > Transport OH > RS

**Syntax**

:SOURCE[1..n]:DATA:TELECOM:SDH:OH:RS:OVERHEAD[1..n] <wsp> <Channel>, A1 | A2  
| J0 | Z0 | B1 | E1 | F1 | D1 | D2 | D3, <Value>

---

## SCPI Command Reference

*OH - SONET/SDH*

---

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]**

<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the regenerator overhead channel number.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: A1   A2   J0   Z0   B1   E1   F1   D1   D2   D3</p> <p>Selects the Regenerator Overhead bytes.</p> <p>A1: F6 hexadecimal value</p> <p>A2: 28 hexadecimal value</p> <p>J0: J0 trace</p> <p>Z0: Growth</p> <p>B1: Bit Interleaved Parity code (BIP-8)</p> <p>E1: Orderwire</p> <p>F1: User</p> <p>D1: Data Communications Channel (DCC)</p> <p>D2: Data Communications Channel (DCC)</p> <p>D3: Data Communications Channel (DCC)</p> <p><b>Value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the regenerator overhead byte values in hexadecimal format.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDH:OH:RS:OVER1 1, A1, #HF6</p> <p>SOUR:DATA:TEL:SDH:OH:RS:OVER1? 1, A1</p> <p>Returns: 246</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:SECTIon:OVERhead?</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]?**

<b>Description</b>	<p>This query returns the regenerator overhead byte values in decimal format.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH &gt; SDH &gt;TX &gt;Transport OH &gt;RS</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]? &lt;wsp&gt; &lt;Channel&gt;, A1   A2   J0   Z0   B1   E1   F1   D1   D2   D3</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the regenerator overhead channel number.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: A1   A2   J0   Z0   B1   E1   F1   D1   D2   D3</p> <p>Selects the Regenerator Section Overhead bytes.</p> <p>A1: F6 hexadecimal value  A2: 28 hexadecimal value  J0: J0 trace  Z0: Growth  B1: Bit Interleaved Parity code (BIP-8)  E1: Orderwire  F1: User  D1: Data Communications Channel (DCC)  D2: Data Communications Channel (DCC)  D3: Data Communications Channel (DCC)</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

## SCPI Command Reference

*OH - SONET/SDH*

---

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the regenerator section overhead byte value for transmitter.
<b>Example(s)</b>	SOUR:DATA:TEL:SDH:OH:RS:OVER1 1, A1, #HF6 SOUR:DATA:TEL:SDH:OH:RS:OVER1? 1, A1 Returns: 246
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:SECTion:OVERhead

---

**:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]:DEFault**

<b>Description</b>	<p>This command resets the Regenerator Section overhead byte values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; OH &gt; SDH &gt; TX &gt; Transport OH &gt; RS &gt; Default</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]:DEFault &lt;wsp&gt;&lt;Channel&gt;, A1   A2   J0   Z0   B1   E1   F1   D1   D2   D3</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the regenerator overhead channel number.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: A1   A2   J0   Z0   B1   E1   F1   D1   D2   D3</p> <p>Selects the Regenerator Section Overhead bytes.</p> <p>A1: F6 hexadecimal value  A2: 28 hexadecimal value  J0: J0 trace  Z0: Growth  B1: Bit Interleaved Parity code (BIP-8)  E1: Orderwire  F1: User  D1: Data Communications Channel (DCC)  D2: Data Communications Channel (DCC)  D3: Data Communications Channel (DCC)</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDH:OH:RS:OVER1:DEF 1, A1</p>
<b>See Also</b>	<p>SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:LINE:OVERhead:DEFault</p>

## SCPI Command Reference

### OH - SONET/SDH

---

**:SENSe[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]?**

<b>Description</b>	<p>This query returns the regenerator overhead byte values in decimal format for the receiver.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH &gt;SDH&gt;Transport OH &gt;RS &gt;RX</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]? &lt;wsp&gt; &lt;Channel&gt;, A1   A2   J0   Z0   B1   E1   F1   D1   D2   D3</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the regenerator overhead channel number for the receiver.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: A1   A2   J0   Z0   B1   E1   F1   D1   D2   D3</p> <p>Selects the Regenerator Overhead bytes for the receiver.</p> <p>A1: F6 hexadecimal value</p> <p>A2: 28 hexadecimal value</p> <p>J0: J0 trace</p> <p>Z0: Growth</p> <p>B1: Bit Interleaved Parity code (BIP-8)</p> <p>E1: Orderwire</p> <p>F1: User</p> <p>D1: Data Communications Channel (DCC)</p> <p>D2: Data Communications Channel (DCC)</p> <p>D3: Data Communications Channel (DCC)</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

---

**:SENSe[1..n]:DATA:TELecom:SDH:OH:RS:OVERhead[1..n]?**

---

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the regenerator overhead byte value for the receiver.
<b>Example(s)</b>	SENS:DATA:TEL:SDH:OH:RS:OVER1? 1, A1
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:SDHSonet:OH:LINE:OVERhead?

---

## SCPI Command Reference

### *OH - SONET/SDH*

---

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]**

**Description**

This command sets the multiplexer overhead byte values in hexadecimal format.

At \*RST condition, this value is device dependent.

Navigation Path: Test Setup >OTN SONET/SDH BERT OR SONET/SDH BERT> Functions > OH> > SDH >TX >Transport OH >MS

**Syntax**

:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n] <wsp> <Channel>, H1 | H2 | H3 | B2 | K1 | K2 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | S1 | M0 | E2 | M1, <Value>

---



---

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]**

<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the multiplexer overhead channel number for transmitter.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   S1   M0   E2   M1</p> <p>Selects the multiplexer overhead bytes.</p> <p>H1 to H3: Respectively H1, H2, or H3 Pointer</p> <p>B2: Bit Interleaved Parity code (BIP-8)</p> <p>K1 and K2: Respectively K1 or K2 Automatic Protection Switching (APS)</p> <p>D4 to D12: Respectively D4 to D12 Data Communications Channel (DCC)</p> <p>S1: Synchronization status</p> <p>M0: Remote Error Indicator - Line (REI-L)</p> <p>M1: Remote Error Indicator - Line (REI-L)</p> <p>E2: Orderwire</p> <p><b>Value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the multiplexer overhead byte values in hexadecimal format.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDH:OH:MS:OVER1 1, H1, #HF6</p> <p>SOUR:DATA:TEL:SDH:OH:MS:OVER1? 1, H1</p> <p>Returns: 246</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:LINE:OVERhead?</p>

---

## SCPI Command Reference

### OH - SONET/SDH

---

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]?**

<b>Description</b>	<p>This query returns the multiplexer overhead byte values in decimal format.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH&gt; &gt; SDH &gt;TX &gt;Transport OH &gt;MS</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]? &lt;wsp&gt; &lt;Channel&gt;, H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   S1   M0   E2   M1</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the multiplexer overhead channel number.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   S1   M0   E2   M1</p> <p>Selects the multiplexer overhead bytes.</p> <p>H1 to H3: Respectively H1, H2, or H3 Pointer</p> <p>B2: Bit Interleaved Parity code (BIP-8)</p> <p>K1 and K2: Respectively K1 or K2 Automatic Protection Switching (APS)</p> <p>D4 to D12: Respectively D4 to D12 Data Communications Channel (DCC)</p> <p>S1: Synchronization status</p> <p>M0: Remote Error Indicator - Line (REI-L)</p> <p>M1: Remote Error Indicator - Line (REI-L)</p> <p>E2: Orderwire</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]?**

---

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the multiplexer overhead byte value.
<b>Example(s)</b>	SOUR:DATA:TEL:SDH:OH:MS:OVER1 1, H1, #HF6 SOUR:DATA:TEL:SDH:OH:MS:OVER1? 1, H1 Returns: 246
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:LINE:OVERhead

---

## SCPI Command Reference

### OH - SONET/SDH

---

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]:DEFault**

<b>Description</b>	<p>This command resets the line overhead byte values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH &gt; SDH &gt;TX &gt;Transport OH &gt;MS &gt;Default</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]:DEFault &lt;wsp&gt; &lt;Channel&gt;, H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   S1   M0   E2</pre>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the multiplexer overhead channel number.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   S1   M0   E2</p> <p>Selects the multiplexer overhead bytes.</p> <p>H1 to H3: Respectively H1, H2, or H3 Pointer</p> <p>B2: Bit Interleaved Parity code (BIP-8)</p> <p>K1 and K2: Respectively K1 or K2 Automatic Protection Switching (APS)</p> <p>D4 to D12: Respectively D4 to D12 Data Communications Channel (DCC)</p> <p>S1: Synchronization status</p> <p>M0: Remote Error Indicator - Line (REI-L)</p> <p>E2: Orderwire</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDH:OH:MS:OVER1:DEF 1, H1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:RS:OVERhead[1..n]:DEFault</pre>

**:SENSe[1..n]:DATA:TELeCom:SDH:OH:MS:OVERhead[1..n]?**

<b>Description</b>	<p>This query returns the multiplexer overhead byte values in decimal format for the receiver.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH &gt;SDH&gt;Transport OH &gt;MS &gt;RX</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELeCom:SDH:OH:MS:OVERhead[1..n]? &lt;wsp&gt; &lt;Channel&gt;, H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   S1   M0   E2   M1</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the multiplexer overhead channel number for the receiver.</p> <p><b>Overhead:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: H1   H2   H3   B2   K1   K2   D4   D5   D6   D7   D8   D9   D10   D11   D12   S1   M0   E2   M1</p> <p>Selects the multiplexer overhead bytes.</p> <p>H1 to H3: Respectively H1, H2, or H3 Pointer</p> <p>B2: Bit Interleaved Parity code (BIP-8)</p> <p>K1 and K2: Respectively K1 or K2 Automatic Protection Switching (APS)</p> <p>D4 to D12: Respectively D4 to D12 Data Communications Channel (DCC)</p> <p>S1: Synchronization status</p> <p>M0: Remote Error Indicator - Line (REI-L)</p> <p>M1: Remote Error Indicator - Line (REI-L)</p> <p>E2: Orderwire</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

## SCPI Command Reference

*OH - SONET/SDH*

---

**:SENSe[1..n]:DATA:TELeom:SDH:OH:MS:OVERhead[1..n]?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the multiplexer overhead byte value for the receiver.
<b>Example(s)</b>	SENS:DATA:TEL:SDH:OH:MS:OVER1? 1, H1
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:SDHSonet:OH:SECTion:OVERhead

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead****Description**

This command sets the High Order Path (HOP) overhead values in hexadecimal format.

At \*RST condition, this value is device dependent.

Navigation Path: Test Setup >OTN SONET/SDH BERT OR SONET/SDH BERT> Functions  
> OH>SONET>TX>STS

Navigation Path: Test Setup >OTN SONET/SDH BERT OR SONET/SDH BERT> Functions  
> OH>SDH>TX>AU

**Syntax**

:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead <wsp>J1 | B3 | C2 | G1 |  
F2 | H4 | Z3 | Z4 | N1 | F3 | K3, <Value>

---

## SCPI Command Reference

*OH - SONET/SDH*

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead**

<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: J1   B3   C2   G1   F2   H4   Z3   Z4   N1   F3   K3</p> <p>Selects the High Order Path (HOP) overhead values.</p> <p>J1: J1 trace</p> <p>B3: Bit Interleaved Parity code (BIP-8)</p> <p>C2: Path Signal Label</p> <p>G1: Path Status</p> <p>F2: User Channel</p> <p>H4: Multiframe Indicator</p> <p>Z3: Growth</p> <p>Z4: Growth</p> <p>N1: Tandem Connection Monitoring</p> <p>F3: Overhead byte</p> <p>K3: Overhead byte</p> <p><b>Value:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the HOP overhead values in hexadecimal format.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:OH:HOP:OVER C2,#H02</p> <p>SOUR:DATA:TEL:SDHS:OH:HOP:OVER? C2</p> <p>Returns: 2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]?</p>



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead?**

<b>Description</b>	<p>This query returns the High Order Path (HOP) overhead values in decimal format.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH&gt;SONET&gt;TX&gt;STS</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH&gt;SDH&gt;TX&gt;AU</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead? <wsp>J1   B3   C2   G1   F2   H4   Z3   Z4   N1   F3   K3
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: J1   B3   C2   G1   F2   H4   Z3   Z4   N1   F3   K3</p> <p>Selects the High Order Path (HOP) overhead values.</p> <p>J1: J1 trace</p> <p>B3: Bit Interleaved Parity code (BIP-8)</p> <p>C2: Path Signal Label</p> <p>G1: Path Status</p> <p>F2: User Channel</p> <p>H4: Multiframe Indicator</p> <p>Z3: Growth</p> <p>Z4: Growth</p> <p>N1: Tandem Connection Monitoring</p> <p>F3: Overhead byte</p> <p>K3: Overhead byte</p>
<b>Response Syntax</b>	<Value>

## SCPI Command Reference

*OH - SONET/SDH*

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the High Order Path (HOP) overhead byte value.
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:OH:HOP:OVER C2,#H02 SOUR:DATA:TEL:SDHS:OH:HOP:OVER? C2 Returns: 2
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]?

---

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead:DEFault**

<b>Description</b>	<p>This command resets the HOP overhead byte values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH&gt;SONET&gt;TX&gt;STS&gt;Default</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt;OH &gt; SDH &gt;TX &gt;AU&gt;Default</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead:DEFault <wsp>J1   B3   C2   G1   F2   H4   Z3   Z4   N1   F3   K3
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: J1   B3   C2   G1   F2   H4   Z3   Z4   N1   F3   K3</p> <p>Selects the High Order Path (HOP) overhead values.</p> <p>J1: J1 trace</p> <p>B3: Bit Interleaved Parity code (BIP-8)</p> <p>C2: Path Signal Label</p> <p>G1: Path Status</p> <p>F2: User Channel</p> <p>H4: Multiframe Indicator</p> <p>Z3: Growth</p> <p>Z4: Growth</p> <p>N1: Tandem Connection Monitoring</p> <p>F3: Overhead byte</p> <p>K3: Overhead byte</p>
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:OH:HOP:OVER:DEF J1
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]:DEFault

## SCPI Command Reference

*OH - SONET/SDH*

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead?**

<b>Description</b>	<p>This query returns the High Order Path (HOP) overhead values in hexadecimal format for the receiver when disabled.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH&gt;SONET&gt;RX&gt;STS</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions &gt; OH&gt;SDH&gt;RX&gt;AU</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead? &lt;wsp&gt;J1   B3   C2   G1   F2   H4   Z3   Z4   N1   F3   K3</p>
<b>Parameter(s)</b>	<p><b>Overhead:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: J1   B3   C2   G1   F2   H4   Z3   Z4   N1   F3   K3</p> <p>Selects the High Order Path (HOP) overhead values for the receiver.</p> <p>J1: J1 trace</p> <p>B3: Bit Interleaved Parity code (BIP-8)</p> <p>C2: Path Signal Label</p> <p>G1: Path Status</p> <p>F2: User Channel</p> <p>H4: Multiframe Indicator</p> <p>Z3: Growth</p> <p>Z4: Growth</p> <p>N1: Tandem Connection Monitoring</p> <p>F3: Overhead byte</p> <p>K3: Overhead byte</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the High Order Path (HOP) overhead byte value for the receiver.
<b>Example(s)</b>	SENS:DATA:TEL:SDHS:OH:HOP:OVER? J1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]

---

## SCPI Command Reference

*OH - SONET/SDH*

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:REStore:DEFault**

<b>Description</b>	<p>This command it restores the default application.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt;Functions &gt;OH&gt;SONET&gt; TX&gt;Default all OH</p> <p>Navigation Path: Test Setup &gt;OTN SONET/SDH BERT OR SONET/SDH BERT&gt; Functions&gt;OH&gt;SDH&gt; TX&gt;Default all OH</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:REStore:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:OH:RES:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:OVERhead[1..n]:DEFault

---

## OH - GFP-F/GFP-T

---

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI

---

<b>Description</b>	<p>This command sets the Type of GFP Client Frame.</p> <p>At *RST condition, this value is 000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Type Header &gt; PTI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Type Header &gt; PTI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI <wsp>DFRames   MFRames, <Pti>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p><b>Pti:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the PTI value between 000 and 111</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:OH:THE:PTI MFR,#B0</p> <p>SOUR:DATA:TEL:GFP:OH:THE:PTI? MFR</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI?

---

## SCPI Command Reference

### OH - GFP-F/GFP-T

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI?**

<b>Description</b>	<p>This query returns the Type of GFP Client Frame.</p> <p>At *RST condition, this value is 000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Type Header &gt; PTI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Type Header &gt; PTI</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI? &lt;wsp&gt;DFRames   MFRames[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current PFI value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Pti&gt;</p>



---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI?**

---

<b>Response(s)</b>	<b>Pti:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Type of GFP Client Frame.
<b>Example(s)</b>	SOUR:DATA:TEL:GFP:OH:THE:PTI MFR,#B0 SOUR:DATA:TEL:GFP:OH:THE:PTI? MFR Returns: 0
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI

---

## SCPI Command Reference

### OH - GFP-F/GFP-T

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI**

<b>Description</b>	<p>This command sets the Payload FCS Indicator.</p> <p>At *RST condition, this value is 0.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Type Header &gt; PFI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Type Header &gt; PFI</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI &lt;wsp&gt;DFRames   MFRames, &lt;Pfi&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p><b>Pfi:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the PFI value between 0 and 1</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:OH:THE:PFI MFR,#B0</p> <p>SOUR:DATA:TEL:GFP:OH:THE:PFI? MFR</p> <p>Returns: 0</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI?**

<b>Description</b>	<p>This query returns the Payload FCS Indicator.</p> <p>At *RST condition, this value is 0.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Type Header &gt; PFI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Type Header &gt; PFI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI? <wsp>DFRames   MFRames[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current PTI value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Pfi>

## SCPI Command Reference

*OH - GFP-F/GFP-T*

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI?**

<b>Response(s)</b>	<b>Pfi:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Payload FCS Indicator.
<b>Example(s)</b>	SOUR:DATA:TEL:GFP:OH:THE:PFI MFR,#B0 SOUR:DATA:TEL:GFP:OH:THE:PFI? MFR Returns: 0
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI

---

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI**


---

<b>Description</b>	<p>This command sets the Extension Header Identifier.</p> <p>At *RST condition, this value is 0000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Type Header &gt; EXI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Type Header &gt; EXI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI <wsp>DFRames   MFRames, <Exi>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p><b>Exi:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the EXI value between 0000 and 1111</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:OH:THE:EXI MFR,#B0</p> <p>SOUR:DATA:TEL:GFP:OH:THE:EXI? MFR</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe?

---

## SCPI Command Reference

### OH - GFP-F/GFP-T

---

**:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI?**

<b>Description</b>	<p>This query returns the value of the Extension Header Identifier.</p> <p>At *RST condition, this value is 0000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Type Header &gt; EXI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Type Header &gt; EXI</p>
<b>Syntax</b>	<p>:SOURCE[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI? &lt;wsp&gt;DFRames   MFRames[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current EXI value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Exi&gt;</p>

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI?**

---

<b>Response(s)</b>	<b>Exi:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Extension Header Identifier value.
<b>Example(s)</b>	SOUR:DATA:TEL:GFP:OH:THE:EXI MFR,#B0 SOUR:DATA:TEL:GFP:OH:THE:EXI? MFR Returns: 0
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI

---

## SCPI Command Reference

### OH - GFP-F/GFP-T

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI**

<b>Description</b>	<p>This command sets the User Payload Identifier.</p> <p>At *RST condition, this value is 0000 0000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Type Header &gt; UPI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Type Header &gt; UPI</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI &lt;wsp&gt;DFRames   MFRames, &lt;Upi&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p><b>Upi:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the UPI value between 0000 0000 and 1111 1111</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:OH:THE:UPI MFR,#B0</p> <p>SOUR:DATA:TEL:GFP:OH:THE:UPI? MFR</p> <p>Returns: 0</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI?</p>



---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI?**

---

<b>Description</b>	<p>This query returns the values of User Payload Identifier.</p> <p>At *RST condition, this value is 0000 0000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Type Header &gt; UPI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Type Header &gt; UPI</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI? <wsp>DFRames   MFRames[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current UPI value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Upi>

---

## SCPI Command Reference

*OH - GFP-F/GFP-T*

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI?**

<b>Response(s)</b>	<b>Upi:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the User Payload Identifier value.
<b>Example(s)</b>	SOUR:DATA:TEL:GFP:OH:THE:UPI MFR,#B0 SOUR:DATA:TEL:GFP:OH:THE:UPI? MFR Returns: 0
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID

---

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID**

---

<b>Description</b>	<p>This command sets the Channel Id.</p> <p>At *RST condition, this value is 0000 0000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Extension Header &gt; CID</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Extension Header &gt; CID</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID <wsp>DFRames   MFRames, <CID>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p><b>CID:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects Channel Id between 0000 0000 to 1111 1111.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:OH:EHE:CID DFR,#B0</p> <p>SOUR:DATA:TEL:GFP:OH:EHE:CID? DFR</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI?

---

## SCPI Command Reference

### OH - GFP-F/GFP-T

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID?**

<b>Description</b>	<p>This query returns the Channel Id.</p> <p>At *RST condition, this value is 0000 0000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Extension Header &gt; CID</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Extension Header &gt; CID</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID? &lt;wsp&gt;DFRames   MFRames[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current CID value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;CID&gt;</p>

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID?**

<b>Response(s)</b>	<b>CID:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Channel Id.
<b>Example(s)</b>	SOUR:DATA:TEL:GFP:OH:EHE:CID DFR,#B0 SOUR:DATA:TEL:GFP:OH:EHE:CID? DFR Returns: 0
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe

---

## SCPI Command Reference

### OH - GFP-F/GFP-T

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe**

<b>Description</b>	<p>This command sets the Extension Header Spare Field.</p> <p>At *RST condition, this value is 0000 0000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Extension Header &gt; Spare</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Extension Header &gt; Spare</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe &lt;wsp&gt;DFRames   MFRames, &lt;Spare&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p><b>Spare:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Spare value between 0000 0000 and 1111 1111</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:OH:EHE:SPAR DFR,#B0</p> <p>SOUR:DATA:TEL:GFP:OH:EHE:SPAR? DFR</p> <p>Returns: 0</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe?**

<b>Description</b>	<p>This query returns the the Extension Header Spare Field value.</p> <p>At *RST condition, this value is 0000 0000.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Extension Header &gt; Spare</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Extension Header &gt; Spare</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe? <wsp>DFRames   MFRames[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRames</p> <p>Selects the Client Framing Types.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current spare value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<SPARe>

## SCPI Command Reference

*OH - GFP-F/GFP-T*

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe?**

<b>Response(s)</b>	<b>SPARe:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the SPARe value.
<b>Example(s)</b>	SOUR:DATA:TEL:GFP:OH:EHE:SPAR DFR,#B0 SOUR:DATA:TEL:GFP:OH:EHE:SPAR? DFR Returns: 0
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID

---



**:FETCh[1..n]:DATA:TELEcom:GFP:OH:DFRames?**

<b>Description</b>	<p>This query returns selection of the Client Data Frames.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; RX &gt; Frames &gt; Client Data</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; RX &gt; Frames &gt; Client Data</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:OH:DFRames? <wsp>PLI   CHEC   PTI   PFI   EXI   UPI   THEC
<b>Parameter(s)</b>	<p><b>Dframes:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PLI   CHEC   PTI   PFI   EXI   UPI   THEC</p> <p>Selects the Client Data Frames.</p> <p>PLI</p> <p>CHEC</p> <p>PTI</p> <p>PFI</p> <p>EXI</p> <p>UPI</p> <p>THEC</p> <p>CID</p> <p>SPARe</p> <p>EHEC</p>
<b>Response Syntax</b>	<Dframes>

## SCPI Command Reference

*OH - GFP-F/GFP-T*

---

**:FETCh[1..n]:DATA:TELEcom:GFP:OH:DFRames?**

<b>Response(s)</b>	<b>Dframes:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the selection of Data Frames.
<b>Example(s)</b>	FETC:DATA:TEL:GFP:OH:DFR? PLI
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI

---

**:FETCh[1..n]:DATA:TELeom:GFP:OH:MFRames?**

<b>Description</b>	<p>This query returns selection of the Client Management Frames.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; RX &gt; Frames &gt; Client Management</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; RX &gt; Frames &gt; Client Management</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELeom:GFP:OH:MFRames? &lt;wsp&gt;PLI   CHEC   PTI   PFI   EXI   UPI   THEC   CID   SPARe   EHEC</p>
<b>Parameter(s)</b>	<p><b>Mframes:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PLI   CHEC   PTI   PFI   EXI   UPI   THEC   CID   SPARe   EHEC</p> <p>Selects the Client Management Frames.</p> <p>PLI</p> <p>CHEC</p> <p>PTI</p> <p>PFI</p> <p>EXI</p> <p>UPI</p> <p>THEC</p> <p>CID</p> <p>SPARe</p> <p>EHEC</p>
<b>Response Syntax</b>	<p>&lt;Mframes&gt;</p>

## SCPI Command Reference

*OH - GFP-F/GFP-T*

---

**:FETCh[1..n]:DATA:TELecom:GFP:OH:MFRames?**

<b>Response(s)</b>	<b>Mframes:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the selection of Management Frames.
<b>Example(s)</b>	FETC:DATA:TEL:GFP:OH:MFRames? PLI
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:GFP:OH:EHEader:SPARe

---

**:FETCh[1..n]:DATA:TELEcom:GFP:OH:RPTiframes?**

<b>Description</b>	<p>This query returns selection of Reserved PTI Frames.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (Ethernet(flex/GFP-F)/10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; RX &gt; Frames &gt; Reserved PTI</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; RX &gt; Frames &gt; Reserved PTI</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:GFP:OH:RPTiframes? <wsp>PLI   CHEC   PTI   PFI   EXI   UPI   THEC   CID   SPARe   EHEC
<b>Parameter(s)</b>	<p><b>Rptiframes:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PLI   CHEC   PTI   PFI   EXI   UPI   THEC   CID   SPARe   EHEC</p> <p>Selects the Reserved PTI Frames.</p> <p>PLI</p> <p>CHEC</p> <p>PTI</p> <p>PFI</p> <p>EXI</p> <p>UPI</p> <p>THEC</p> <p>CID</p> <p>SPARe</p> <p>EHEC</p>
<b>Response Syntax</b>	<Rptiframes>

## SCPI Command Reference

*OH - GFP-F/GFP-T*

---

---

**:FETCh[1..n]:DATA:TELEcom:GFP:OH:RPTiframes?**

<b>Response(s)</b>	<b>Rptiframes:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the selection of Reserved PTI Frames.
<b>Example(s)</b>	FETC:DATA:TEL:GFP:OH:RPT? PLI
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:EHeader:CID

---

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:REStore:DEFault**

---

<b>Description</b>	<p>This command resets or overwrites the overhead byte values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Default All OH</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Default All OH</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:GFP:OH:REStore:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:GFP:OH:REStore:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:GFP:OH:EHeader:SPARe

---

## SCPI Command Reference

OH - GFP-F/GFP-T

---

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:DEFault**

<b>Description</b>	<p>This command resets or overwrites the overhead byte values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (10GbE) &gt; GFP-F &gt; Functions &gt; OH &gt; GFP-F &gt; Default</p> <p>Navigation Path: Test Setup &gt; OTN BERT &gt; Test Configurator &gt; Modify Structure &gt; Client (1GbE) &gt; GFP-T &gt; Functions &gt; OH &gt; GFP-T &gt; Default</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:GFP:OH:DEFault &lt;wsp&gt;DFRames   MFRame, PTI   PFI   EXI   UPI   CID   SPARe</p>
<b>Parameter(s)</b>	<p><b>Frames:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: DFRames   MFRame</p> <p>Selects the Frame Type.</p> <p>DFRames: DFRames</p> <p>MFRames: MFRames</p> <p><b>Bytes:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PTI   PFI   EXI   UPI   CID   SPARe</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:GFP:OH:DEF DFR,PTI</p>
<b>See Also</b>	<p>SOURce:DATA:TELEcom:GFP:OH:RESStore:DEFault</p>



## RTD

---

:SENSe[1..n]:DATA:TELEcom:RTD:MODE

---

<b>Description</b>	<p>This command selects the round trip delay test mode.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: OTN BERT &gt; Functions &gt; RTD</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:RTD:MODE <wsp>SINGle   CONTInuous
<b>Parameter(s)</b>	<p><b>Mode:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SINGle   CONTInuous</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:RTD:MODE SING</p> <p>SENS:DATA:TEL:RTD:MODE?</p> <p>Returns: SINGLE</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:RTD</p> <p>SENSe[1..n]:DATA:TELEcom:RTD?</p>

---

## SCPI Command Reference

RTD

---

:SENSe[1..n]:DATA:TELEcom:RTD:MODE?

<b>Description</b>	<p>This query returns the round trip delay test mode.</p> <p>At *RST condition, this value is set to SINGLE.</p> <p>Navigation Path: OTN BERT &gt; Functions &gt; RTD</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:RTD:MODE?
<b>Response Syntax</b>	<Mode>
<b>Response(s)</b>	<p><b>Mode:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the round trip delay test mode.</p> <p>SINGLE, single mode is selected as the round trip delay.</p> <p>CONTINUOUS, continuous mode is selected as the round trip delay.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:RTD:MODE CONT</p> <p>SENS:DATA:TEL:RTD:MODE?</p> <p>Returns: SINGLE</p>
<b>Note(s)</b>	<CHARACTER RESPONSE DATA>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:RTD</p> <p>SENSe[1..n]:DATA:TELEcom:RTD?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:RTD**

<b>Description</b>	<p>This command enables or disables the round trip delay measurements.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: OTN BERT &gt; Functions &gt; RTD &gt; Measure Delay</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:RTD &lt;wsp&gt;ON   OFF</code>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:RTD:MODE CONT SENS:DATA:TEL:RTD ON SENS:DATA:TEL:RTD? Returns: 1</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:RTD:MODE SENSe[1..n]:DATA:TELEcom:RTD:MODE?</pre>

---

## SCPI Command Reference

### RTD

---

:SENSe[1..n]:DATA:TELEcom:RTD?

<b>Description</b>	<p>This query returns the status of round trip delay measurements.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: OTN BERT &gt; Functions &gt; RTD &gt; Measure Delay</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:RTD?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the round trip delay measurement.</p> <p>1, enables the round trip delay.</p> <p>0, disables the round trip delay.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:RTD:MODE CONT</p> <p>SENS:DATA:TEL:RTD ON</p> <p>SENS:DATA:TEL:RTD?</p> <p>Returns: 1</p>
<b>Note(s)</b>	<NR1 NUMERIC RESPONSE DATA>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:RTD:MODE</p> <p>SENSe[1..n]:DATA:TELEcom:RTD:MODE?</p>

---

**:FETCh[1..n]:DATA:TELEcom:RTD:DELay:STATus?**

---

<b>Description</b>	<p>This query returns the test status of the Round Trip Delay measurements.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: OTN BERT &gt; Functions &gt; RTD</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:RTD:DELay:STATus?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the test status of the Round Trip Delay measurements.</p> <p>READY, Ready indicates that the test is ready to perform RTD measurement.</p> <p>RUNNING, Running indicates that the RTD test is running.</p> <p>CANCELLED, Cancelled indicates that the RTD test has been stopped before its completion.</p> <p>CFAILED, Calibration Failed indicates that the test calibration failed.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:RTD:MODE CONT SENS:DATA:TEL:RTD ON FETC:DATA:TEL:RTD:DEL:STAT?</pre>
<b>Note(s)</b>	<CHARACTER RESPONSE DATA>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:RTD:DELay:LAST?

---

## SCPI Command Reference

### RTD

---

**:FETCh[1..n]:DATA:TELEcom:RTD:DELAy:LAST?**

<b>Description</b>	This query returns the result of the last Round Trip Delay. At *RST condition, this value is set to device-dependent. Navigation Path: OTN BERT > Functions > RTD
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:RTD:DELAy:LAST?
<b>Response Syntax</b>	<Last>
<b>Response(s)</b>	<b>Last:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the result of the last Round Trip Delay.
<b>Example(s)</b>	SENS:DATA:TEL:RTD:MODE SING SENS:DATA:TEL:RTD ON FETC:DATA:TEL:RTD:DEL:LAST?
<b>Note(s)</b>	<STRING RESPONSE DATA>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:RTD:DELAy:MAXimum?

---

---

**:FETCh[1..n]:DATA:TELEcom:RTD:DELay:MAXimum?**

---

<b>Description</b>	This query returns the maximum Round Trip Delay recorded. At *RST condition, this value is set to device-dependent. Navigation Path: OTN BERT > Functions > RTD
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:RTD:DELay:MAXimum?
<b>Response Syntax</b>	<Maximum>
<b>Response(s)</b>	<b>Maximum:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the maximum Round Trip Delay recorded.
<b>Example(s)</b>	SENS:DATA:TEL:RTD:MODE SING SENS:DATA:TEL:RTD ON FETC:DATA:TEL:RTD:DEL:MAX?
<b>Note(s)</b>	<STRING RESPONSE DATA>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:RTD:DELay:LAST?

---

## SCPI Command Reference

### RTD

---

---

#### :FETCh[1..n]:DATA:TELEcom:RTD:DELay:MINimum?

---

<b>Description</b>	This query returns the minimum Round Trip Delay recorded. At *RST condition, this value is set to device-dependent. Navigation Path: OTN BERT > Functions > RTD
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:RTD:DELay:MINimum?
<b>Response Syntax</b>	<Minimum>
<b>Response(s)</b>	<b>Minimum:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the minimum Round Trip Delay recorded.
<b>Example(s)</b>	SENS:DATA:TEL:RTD:MODE SING SENS:DATA:TEL:RTD ON FETC:DATA:TEL:RTD:DEL:MIN?
<b>Note(s)</b>	<STRING RESPONSE DATA>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:RTD:DELay:MAXimum?

---



**:FETCh[1..n]:DATA:TELEcom:RTD:DELay:AVERage?**

<b>Description</b>	<p>This query returns the average Round Trip Delay value.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: OTN BERT &gt; Functions &gt; RTD</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:RTD:DELay:AVERage?
<b>Response Syntax</b>	<Average>
<b>Response(s)</b>	<p><b>Average:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the average Round Trip Delay value.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:RTD:MODE SING</p> <p>SENS:DATA:TEL:RTD ON</p> <p>FETC:DATA:TEL:RTD:DEL:AVER?</p>
<b>Note(s)</b>	<STRING RESPONSE DATA>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:RTD:DELay:MINimum?

---

## SCPI Command Reference

### RTD

---

---

#### :FETCh[1..n]:DATA:TELEcom:RTD:COUNt:SUCCEssful?

---

<b>Description</b>	This query returns the total number of successful measurements. At *RST condition, this value is set to device-dependent. Navigation Path: OTN BERT > Functions > RTD
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:RTD:COUNt:SUCCEssful?
<b>Response Syntax</b>	<Successful>
<b>Response(s)</b>	<b>Successful:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the total number of successful measurements.
<b>Example(s)</b>	SENS:DATA:TEL:RTD:MODE SING SENS:DATA:TEL:RTD ON FETC:DATA:TEL:RTD:COUN:SUC?
<b>Note(s)</b>	<NR2 NUMERIC RESPONSE DATA>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:RTD:COUNt:FAILed?

---

---

**:FETCh[1..n]:DATA:TELEcom:RTD:COUNt:FAILEd?**

<b>Description</b>	This query returns the total number of failed measurements. At *RST condition, this value is set to device-dependent. Navigation Path: OTN BERT > Functions > RTD
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:RTD:COUNt:FAILEd?
<b>Response Syntax</b>	<Failed>
<b>Response(s)</b>	<b>Failed:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the total number of failed measurements.
<b>Example(s)</b>	SENS:DATA:TEL:RTD:MODE SING SENS:DATA:TEL:RTD ON FETC:DATA:TEL:RTD:COUN:FAIL?
<b>Note(s)</b>	<NR2 NUMERIC RESPONSE DATA>
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:RTD:COUNt:SUCCEssful?

---

## SCPI Command Reference

### RTD

---

**:SENSe[1..n]:DATA:TELecom:RTD:RESet**

<b>Description</b>	<p>This command resets the results and measurement count.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: OTN BERT &gt; Functions &gt; RTD</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:RTD:RESet
<b>Example(s)</b>	<p>SENS:DATA:TEL:RTD:MODE SING</p> <p>SENS:DATA:TEL:RTD ON</p> <p>SENS:DATA:TEL:RTD:RES</p>
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:RTD:COUNT:SUCCessful?

## Pointer Adjustment

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:VALue?**

---

<b>Description</b>	<p>This query returns the actual pointer value being transmitted into the SONET/SDH for High Order Path (HOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Step &gt; Pointer Value</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:VALue?
<b>Response Syntax</b>	<Size>
<b>Response(s)</b>	<p><b>Size:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the actual pointer value being transmitted.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:HOP:POIN:VAL?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SIZE?

---

## SCPI Command Reference

### *Pointer Adjustment*

---

<b>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SIZE</b>	
<b>Description</b>	<p>This command selects the number of positive pointer adjustment to include into the SONET or SDH for High Order Path (HOP).</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Step &gt; Value</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SIZE &lt;wsp&gt; &lt;Size&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Size:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the pointer size between 1 through 1000.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:HOP:POIN:INCR:SIZE 15 SOUR:DATA:TEL:SDHS:HOP:POIN:INCR:SIZE? Returns: 15</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SIZE?</code>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SIZE?**

<b>Description</b>	<p>This query returns the selected number of positive pointer adjustment to include into the SONET or SDH for High Order Path (HOP).</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Step &gt; Value</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SIZE?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current number of positive pointer is returned.</p> <p>MAXimum: Biggest supported value MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Size>
<b>Response(s)</b>	<p><b>Size:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the pointer size.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:POIN:INCR:SIZE 15</p> <p>SOUR:DATA:TEL:SDHS:HOP:POIN:INCR:SIZE?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SIZE

## SCPI Command Reference

### *Pointer Adjustment*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SIZE**

---

<b>Description</b>	<p>This command selects the number of negative pointer adjustment to include into the SONET or SDH for High Order Path (HOP).</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Step &gt; Value</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SIZE &lt;wsp&gt; &lt;Size&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Size:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the pointer size between 1 through 1000.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:HOP:POIN:DECR:SIZE 15 SOUR:DATA:TEL:SDHS:HOP:POIN:DECR:SIZE? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SIZE?</pre>

---



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SIZE?**

<b>Description</b>	<p>This query returns the selected number of negative pointer adjustment to include into the SONET or SDH for High Order Path (HOP).</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Step &gt; Value</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SIZE? [<wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current number of negative pointer is returned.</p>
<b>Response Syntax</b>	<Size>
<b>Response(s)</b>	<p><b>Size:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the pointer size.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:POIN:DECR:SIZE 15</p> <p>SOUR:DATA:TEL:SDHS:HOP:POIN:DECR:SIZE?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SIZE

## SCPI Command Reference

### *Pointer Adjustment*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement**

---

<b>Description</b>	<p>This command selects the new increment pointer value for High Order Path (HOP).</p> <p>This command is an event and has no associated *RST condition or query form.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Step &gt; Increment</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:POIN:INCR:SIZE 15</p> <p>SOUR:DATA:TEL:SDHS:HOP:POIN:INCR</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SIZE

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement**

<b>Description</b>	<p>This command selects the new decrement pointer value for High Order Path (HOP).</p> <p>This command is an event and has no associated *RST condition or query form.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Step &gt; Decrement</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:POIN:DECR:SIZE 15</p> <p>SOUR:DATA:TEL:SDHS:HOP:POIN:DECR</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SIZE

---

## SCPI Command Reference

### *Pointer Adjustment*

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue**

<b>Description</b>	<p>This command sets the new pointer value for SONET/SDH of High Order Path (HOP). At *RST condition, this value reverts the default value.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Jump &gt; New Pointer</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue &lt;wsp&gt; &lt;Value&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Value:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element. The allowed elements for this parameter are: MAXimum   MINimum Selects the new pointer value between 0 through 782. MAXimum: Biggest supported value MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:VAL 15 SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:VAL? Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TEL:SDHSonet:HOP:POINter:NEW</p>

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue?**

<b>Description</b>	<p>This query returns the new pointer value for High Order Path (HOP).</p> <p>At *RST condition, this value reverts the default value.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Jump &gt; New Pointer</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current value of new pointer is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the new pointer value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:VAL 15</p> <p>SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:VAL?</p> <p>Returns: 15</p>
<b>See Also</b>	SOURce[1..n]:DATA:TEL:SDHSonet:HOP:POINter:NEW?

## SCPI Command Reference

### *Pointer Adjustment*

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW**

<b>Description</b>	<p>This event sets the new pointer value for High Order Path (HOP).</p> <p>This command is an event and has no associated *RST condition or query form.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Jump &gt; New Pointer &gt; Inject</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:VAL 15</p> <p>SOUR:DATA:TEL:SDHS:HOP:POIN:NEW</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:FLAG**

<b>Description</b>	<p>This command selects the status of a new pointer data flag for High Order Path (HOP).</p> <p>At *RST condition, this value is set to NNDF.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Jump &gt; New Data Flag</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:FLAG &lt;wsp&gt;NNDF   NDF</code>
<b>Parameter(s)</b>	<p><b>Flag:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NNDF   NDF</p> <p>Selects the new pointer data flag.</p> <p>NNDF: the NNDF (No New Data Flag).</p> <p>NDF: the NDF (New Data Flag).</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:FLAG NNDF SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:FLAG? Returns: NNDF</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue?</code>

---

## SCPI Command Reference

### *Pointer Adjustment*

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:FLAG?**

<b>Description</b>	<p>This query returns the status of a new pointer data flag for High Order Path (HOP). At *RST condition, this value is set to NNDF. Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Jump &gt; New Data Flag</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:FLAG?
<b>Response Syntax</b>	<Flag>
<b>Response(s)</b>	<p><b>Flag:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the new pointer data flag.</p> <p>NNDF, No New Data Flag (NNDF) is selected.</p> <p>NDF, New Data Flag (NDF) is selected.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:FLAG NNDF SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:FLAG? Returns: NNDF</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue

---



**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:VALue?**

<b>Description</b>	<p>This query returns the actual pointer value being transmitted for High Order Path (HOP). At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; RX Pointer Adjustment &gt; Pointer Value</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:VALue?
<b>Response Syntax</b>	<Pointer>
<b>Response(s)</b>	<p><b>Pointer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the actual pointer value being transmitted.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:POIN:VAL?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue

---

## SCPI Command Reference

### *Pointer Adjustment*

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:OFFSet?**

<b>Description</b>	<p>This query returns the difference between the pointer increment and the pointer decrement for High Order Path (HOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; RX Pointer Adjustment &gt; Cumulative Offset</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:OFFSet?
<b>Response Syntax</b>	<Offset>
<b>Response(s)</b>	<p><b>Offset:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the difference between the pointer increment and the pointer decrement.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:POIN:OFFS?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:COUNT?**

<b>Description</b>	<p>This query returns the count in which positive pointer adjustment is detected for High Order Path (HOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; RX Pointer Adjustment &gt; Ptr.Incr. Count</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:COUNT?
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of positive pointer adjustment detected.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:POIN:INCR:COUN?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement

---

## SCPI Command Reference

### *Pointer Adjustment*

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:COUNT?**

<b>Description</b>	<p>This query returns the counts in which negative pointer adjustment is detected for High Order Path (HOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; RX Pointer Adjustment &gt; Ptr.Decr. Count</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:COUNT?
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the count of negative pointer adjustment detected.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:POIN:DECR:COUN?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement

---

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SEConds?**

---

<b>Description</b>	<p>This query returns the number of seconds in which positive pointer adjustment is detected for High Order Path (HOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; RX Pointer Adjustment &gt; Ptr.Incr. Seconds</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement:SEConds?
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of positive pointer adjustment detected.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:POIN:INCR:SEC?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:INCRement

---

## SCPI Command Reference

### *Pointer Adjustment*

---

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds in which negative pointer adjustment is detected for High Order Path (HOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; RX Pointer Adjustment &gt; Ptr.Decr. Seconds</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement:SEConds?
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of negative pointer adjustment detected.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:POIN:DECR:SEC?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:DECRement

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NDF:COUNT?**

<b>Description</b>	<p>This query returns the count of New Data Flag (NDF).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; RX Pointer Adjustment &gt; NDF Count</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NDF:COUNT?
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of counts of New Data Flag (NDF).</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:POIN:NDF:COUN?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:FLAG

---

## SCPI Command Reference

### *Pointer Adjustment*

---

---

#### **:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NNDF:COUNT?**

---

<b>Description</b>	<p>This query returns the count of No New Data Flag (NNDF).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; RX Pointer Adjustment &gt; No NDF Count</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NNDF:COUNT?
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of counts of No New Data Flag (NNDF).</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:POIN:NNDF:COUN?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:FLAG

---



**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NDF:SEConds?**

<b>Description</b>	<p>This query returns the number of seconds of New Data Flag (NDF) for High Order Path (HOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; RX Pointer Adjustment &gt; NDF Seconds</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NDF:SEConds?
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of New Data Flag (NDF).</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:POIN:NDF:SEC?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:FLAG

---

## SCPI Command Reference

### *Pointer Adjustment*

---

---

#### **:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NNDF:SEConds?**

---

<b>Description</b>	<p>This query returns the number of seconds of No New Data Flag (NNDF) for High Order Path (HOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt; OTN SONET/SDH BERT OR SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; RX Pointer Adjustment &gt; No NDF Seconds</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NNDF:SEConds?
<b>Response Syntax</b>	<Seconds>
<b>Response(s)</b>	<p><b>Seconds:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of seconds of No New Data Flag ( NNDF).</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:HOP:POIN:NNDF:SEC?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:FLAG

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:PATtern**

<b>Description</b>	<p>This command sets the Pattern type for pointer adjustment to include into the SONET/SDH BERT.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt;Sequence</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:PATtern &lt;wsp&gt;OPPOSITEPOL   DOUBLEPOINTER   MISPOINTER   DOUBLEOPPOLARITY   SINGLEPA   BURSTPA   PHASETRANS   PER873   ADDPER873   CANCELPER873   PERCONTINIOUS   ADDPERCONTINIOUS   CANCELPERCONTINIOUS   PER261   ADDPER261   CANCELPER261</p>

---

## SCPI Command Reference

### Pointer Adjustment

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEquence:POINter:PATtern**

<b>Parameter(s)</b>	<b>Standard:</b>
	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: OPPOSITEPOL   DOUBLEPOINTER   MISPOINTER   DOUBLEOPPOLARITY   SINGLEPA   BURSTPA   PHASETRANS   PER873   ADDPER873   CANCELPER873   PERCONTINUOUS   ADDPERCONTINUOUS   CANCELPERCONTINUOUS   PER261   ADDPER261   CANCELPER261</p> <p>Selects the Pattern type for pointer adjustment.</p> <p>OPPOSITEPOLARITY: Single pointers of opposite polarity.</p> <p>DOUBLEPOINTER: Regular pointers plus one double pointer.</p> <p>MISPOINTER: Regular pointers with one missing pointer.</p> <p>DOUBLEOPPOLARITY: Double pointers of opposite polarity.</p> <p>SINGLEPA: Double pointers of opposite polarity.</p> <p>BURSTPA: Burst pointer adjustment.</p> <p>PHASETRANS: Phase transient.</p> <p>PAPATTERN: Periodic pointer adjustment 87-3 pattern.</p> <p>ADD87: Periodic 87-3 with Add.</p> <p>CANCEL87: Periodic 87-3 with Cancel.</p> <p>PACONTINUOUS: Periodic pointer adjustment continuous.</p> <p>ADDPACONTINUOUS: Periodic pointer adjustment continuous with Add.</p> <p>CANCELPACONTINUOUS: Periodic pointer adjustment continuous with Cancel.</p> <p>PER261: Periodic pointer adjustment 26-1 pattern.</p> <p>ADDPER261: Periodic 26-1 with Add.</p> <p>CANCELPER261: Periodic 26-1 with Cancel.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:SEquence:POINter:PATtern PHASETRANS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:NEW:FLAG?

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:PATtern?**

---

<b>Description</b>	<p>This query returns the Pattern type for pointer adjustment to include into the SONET/SDH BERT.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt;Sequence</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:PATtern?
<b>Response Syntax</b>	<Pattern>

---

## SCPI Command Reference

### Pointer Adjustment

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:PATtern?**

<b>Response(s)</b>	<b>Pattern:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the Pattern type for pointer adjustment. OPPOSITEPOLARITY, Single pointers of opposite polarity is selected. DOUBLEPOINTER, Regular pointers plus one double pointer is selected. MISPOINTER, Regular pointers with one missing pointer is selected. DOUBLEOPPOLARITY, Double pointers of opposite polarity is selected. SINGLEPA, Double pointers of opposite polarity is selected. BURSTPA, Burst pointer adjustment is selected. PHASETRANS, Phase transient is selected. PAPATTERN, Periodic pointer adjustment 87-3 pattern is selected. ADD87, Periodic 87-3 with Add is selected. CANCEL87, Periodic 87-3 with Cancel is selected. PACONTINIOUS, Periodic pointer adjustment continuous is selected. ADDPACONTINIOUS, Periodic pointer adjustment continuous with Add is selected. CANCELPACONTINIOUS, Periodic pointer adjustment continuous with Cancel is selected. PER261, Periodic pointer adjustment 26-1 pattern is selected. ADDPER261, Periodic 26-1 with Add is selected. CANCELPER261, Periodic 26-1 with Cancel is selected.
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:SEQuence:POINter:PATtern? Returns: PHASETRANS
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:NEW:FLAG

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:TYPE**

<b>Description</b>	<p>This command sets the Pointer type for pointer adjustment to include into the SONET/SDH BERT.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt;Sequence</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:TYPE <wsp>INCR   DECR
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: INCR   DECR</p> <p>Selects the Pointer type for pointer adjustment.</p> <p>INCR: Increment</p> <p>DECR: Decrement</p>
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:SEQuence:POINter:TYPE INCR
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue

---

## SCPI Command Reference

### Pointer Adjustment

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:TYPE?**

<b>Description</b>	<p>This query returns the Pointer type for pointer adjustment to include into the SONET/SDH BERT.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt;Sequence</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:TYPE?
<b>Response Syntax</b>	<Pointer>
<b>Response(s)</b>	<p><b>Pointer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Pointer type for pointer adjustment.</p> <p>INCR, INCR is selected as Pointer type for pointer adjustment</p> <p>DECR, DECR is selected as Pointer type for pointer adjustment</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:SEQuence:POINter:TYPE?</p> <p>Returns: INCR</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:NEW:VALue

---



**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:PERiodic:STATus**

<b>Description</b>	<p>This command sets the actual pointer periodic status being transmitted for Low Order Path (LOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt;Periodic</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:PERiodic:STATus &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the actual pointer periodic status for the Low Order Path (LOP) error.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:SEQuence:POINter:PERiodic:STATus ON SOUR:DATA:TEL:SDHS:SEQuence:POINter:PERiodic:STATus? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:INCRement</pre>

---

## SCPI Command Reference

### *Pointer Adjustment*

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:PERiodic:STATus?**

<b>Description</b>	<p>This query returns the actual pointer periodic status being transmitted for Low Order Path (LOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt;Periodic</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:PERiodic:STATus?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of actual pointer periodic status.</p> <p>1, actual pointer periodic status is enabled.</p> <p>0, actual pointer periodic status is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:SEQuence:POINter:PERiodic:STATUS?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:DECRement

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:INITcool:STATus**

---

<b>Description</b>	<p>This command sets the actual pointer init-cool status being transmitted for Low Order Path (LOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt;Init-Cool</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:INITcool:STATus &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the actual pointer init-cool status for the Low Order Path (LOP) error.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:SDHS:SEQuence:POINter:INITcool:STATUS ON SOUR:DATA:TEL:SDHS:SEQuence:POINter:INITcool:STATUS? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:INCRement</pre>

---

## SCPI Command Reference

### Pointer Adjustment

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:INITcool:STATUS?**

<b>Description</b>	<p>This query returns the actual pointer init-cool status being transmitted for Low Order Path (LOP).</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt;Init-Cool</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter:INITcool:STATUS?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the actual pointer init-cool status of Low Order Path (LOP) alarm generation.</p> <p>1, actual pointer init-cool status is enabled.</p> <p>0, actual pointer init-cool status is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:SEQUence:POINter:INITcool:STATUS?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:DECREment

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:TIMeline:VALue**

<b>Description</b>	<p>This command sets the Timeline Value status being transmitted for Low Order Path (LOP). At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt;T1(s)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:TIMeline:VALue <wsp>T1   T2   T3   T4   T5   T6, <Size>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: T1   T2   T3   T4   T5   T6</p> <p>Selects the Timeline value for pointer adjustment.</p> <p>T1: T1. T2: T2. T3: T3. T4: T4. T5: T5. T6: T6.</p> <p><b>Size:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Timeline value being transmitted for Low Order Path (LOP). Choices are 10 through 30</p> <p>MAXimum: Biggest supported value MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:SEQuence:POINter:TIMeline:VALue T1, 10 SOUR:DATA:TEL:SDHS:SEQuence:POINter:TIMeline:VALue? Returns: 10</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:NEW:FLAG

---

## SCPI Command Reference

### Pointer Adjustment

---

**:SOURce[1..n]:DATA:TELEcom:SDHSONet:SEQuence:POINter:TIMeline:VALue?**

<b>Description</b>	<p>This query returns the Timeline Value status being transmitted for Low Order Path (LOP). At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt;T1(s)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:SDHSONet:SEQuence:POINter:TIMeline:VALue? &lt;wsp&gt;T1   T2   T3   T4   T5   T6[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: T1   T2   T3   T4   T5   T6</p> <p>Selects the Timeline value for pointer adjustment.</p> <p>T1: T1. T2: T2. T3: T3. T4: T4. T5: T5. T6: T6.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the Timeline value is returned.</p> <p>MAXimum: Biggest supported value MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:TIMeline:VALue?**

---

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Timeline value being transmitted for Low Order Path (LOP).
<b>Example(s)</b>	SOUR:DATA:TEL:SDHS:SEQuence:POINter:TIMeline:VALue? Returns: 10
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:NEW:FLAG

---

## SCPI Command Reference

### *Pointer Adjustment*

---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter**

<b>Description</b>	<p>This command enables or disables the sequence pointer.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt; Sequence</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the the sequence pointer.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:SEQuence:POINter ON</p> <p>SOUR:DATA:TEL:SDHS:SEQuence:POINter?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:NEW:FLAG

---



**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter?**

<b>Description</b>	<p>This query returns the status of the sequence pointer.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; SONET/SDH BERT &gt; Functions &gt; Pointer Adjustment &gt; TX Pointer Adjustment &gt; Sequence</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:SEQUence:POINter?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the sequence pointer.</p> <p>1, sequence pointer is enabled.</p> <p>0, sequence pointer is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:SDHS:SEQUence:POINter ON</p> <p>SOUR:DATA:TEL:SDHS:SEQUence:POINter?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:NEW:FLAG

---

## SCPI Command Reference

### *Pointer Adjustment*

---

---

#### **:FETCh[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:VALue?**

---

<b>Description</b>	<p>This query returns the actual pointer value being transmitted for Low Order Path (LOP). At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt; Pointer Value</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:VALue?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the sequence pointer value.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:SEQuence:POINter:VALue?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:NEW:FLAG

---

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:STATus?**

<b>Description</b>	<p>This query returns the actual pointer status being transmitted for Low Order Path (LOP). At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test Setup &gt;SONET/SDH BERT&gt; Functions &gt;Pointer Adjustment &gt; TX Pointer Adjustment &gt; Status</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:SDHSonet:SEQuence:POINter:STATus?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the sequence pointer.</p> <p>1, sequence pointer status is enabled. 0, sequence pointer status is disabled.</p>
<b>Example(s)</b>	FETC:DATA:TEL:SDHS:SEQuence:POINter:STATus?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:DECREment

---

## 40/100G Advanced - Lanes Mapping & Skew

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:RESet**

---

<b>Description</b>	<p>This command resets the SKEW (Bit Time) for the link layer.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Reset Skew (dropdown)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:RESet
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:LLAY:SKEW:RES
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPing:DEFault

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:DEFault**

---

<b>Description</b>	<p>This command injects the default mapping for the link layer.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Default Mapping (dropdown)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:LLAY:MAPP:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:MANual

---

## SCPI Command Reference

### *40/100G Advanced - Lanes Mapping & Skew*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:RANDom**

<b>Description</b>	<p>This command injects the random mapping for the link layer.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Random Mapping (dropdown)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:RANDOM
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:LLAY:MAPP:RAND
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:DEFAULT

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:MANual**

---

<b>Description</b>	<p>This command selects the Manual Mapping mode for the link layer.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Mapping (dropdown)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:MANual
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:LLAY:MAPP:MAN
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:MAPPING:RANDom

---

## SCPI Command Reference

### 40/100G Advanced - Lanes Mapping & Skew

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:THRshold**

<b>Description</b>	<p>This command sets the OTL Threshold Value.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: OTN BERT Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Skew Alarms Threshold</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:THRshold &lt;wsp&gt; &lt;Threshold&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Threshold:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets Skew Threshold.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTL:THR 20</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THRshold</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THRshold?</p>

---



**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:THRshold?**

<b>Description</b>	<p>This query returns the Alarm Threshold Value for TX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: OTN BERT Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Skew Alarms Threshold</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:THRshold?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets Skew Threshold.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Skew Alarm Thershold>
<b>Response(s)</b>	<p><b>Skew Alarm Thershold:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Skew Threshold.</p> <p>MAXimum, selects maximum as the Skew Threshold.</p> <p>MINimum, selects minimum as the Skew Threshold.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTL:THR 20</p> <p>SOUR:DATA:TEL:OTN:OTL:THR?</p> <p>Returns: 20</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THRshold</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:THRshold?</p>

## SCPI Command Reference

### *40/100G Advanced - Lanes Mapping & Skew*

---

<b>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:THReshold:DEFault</b>	
<b>Description</b>	<p>This command resets the OTL Threshold Value to its default value.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: OTN BERT Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Skew Alarms Threshold Default</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:THReshold:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:THR:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:RESet

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:RESet**

---

<b>Description</b>	<p>This command resets the SKEW (Bit Time) for the link layer.</p> <p>At *RST condition, this value is set to Reset Skew.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Reset Skew (dropdown)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:RESet
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:SKEW:RES
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTL:THReshold:DEFault

---

## SCPI Command Reference

### *40/100G Advanced - Lanes Mapping & Skew*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:DEFault**

<b>Description</b>	This command injects the default mapping for the link layer. At *RST condition, this value is set to Default Mapping. Navigation Path: OTN BERT Test > Functions > 40/100G Advanced > Lanes Mapping & Skew > Mapping
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:DEFault
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:MAPP:DEF
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:RANDom

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:RANDOM**

---

<b>Description</b>	<p>This command injects the random mapping for the link layer.</p> <p>At *RST condition, this value is set to Default Mapping.</p> <p>Navigation Path: OTN BERT Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Mapping</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:RANDOM
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:MAPP:RAND
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:MANual

---

## SCPI Command Reference

### *40/100G Advanced - Lanes Mapping & Skew*

---

---

#### **:SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:MANual**

---

<b>Description</b>	<p>This command selects the Manual Mapping mode for link layer.</p> <p>At *RST condition, this value is set to Default Mapping.</p> <p>Navigation Path: OTN BERT Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Mapping</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:MANual
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:MAPP:MAN
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTL:MAPPING:RANDom

---

**:SENSe[1..n]:DATA:TELecom:OTN:OTL:RX?**

<b>Description</b>	<p>This query returns the value of the received bits which depends on the selected link layer.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: OTN BERT Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew RX</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:OTN:OTL:RX? <wsp> <Physical Lane>, LOGicallane   SKEW
<b>Parameter(s)</b>	<p><b>Physical Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the physical lane skew. The range for the physical lane is from 0 to 19.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOGicallane   SKEW</p> <p>Selects the Link Layer Type.</p> <p>SKEW: SKEW</p> <p>LOGicallane: Logical lane</p>
<b>Response Syntax</b>	<Llayer>
<b>Response(s)</b>	<p><b>Llayer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the link layer virtual to physical lane skew for the receiver.</p>
<b>Example(s)</b>	SENS:DATA:TEL:OTN:OTL:RX? 1, SKEW
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:OTN:OTL:TX?

## 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

:SOURce[1..n]:DATA:TELEcom:CFP:CSETting

<b>Description</b>	<p>This command enables/disables Control settings for TX &amp; RX IC RST, TX Disable, Module Low Power Mode, Module Reset, CFP Power Shutdown.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; CFP Control Pins</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:CFP:CSETting &lt;wsp&gt;TXRXRST   TXDis   MOLOpwn   MODRst   CFPshutdown, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>State:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TXRXRST   TXDis   MOLOpwn   MODRst   CFPshutdown</p> <p>Selects the Control settings.</p> <p>TXRXRST: TX &amp; RX IC RST.</p> <p>TXDis: TX Disable.</p> <p>MOLOpwn: Module Low Power Mode.</p> <p>MODRst: Module Reset.</p> <p>CFPshutdown: CFP Power Shutdown.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the selected control settings.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CFP:CSET CFP,ON</p> <p>SOUR:DATA:TEL:CFP:CSET? CFP</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:CFP:STATus?</p>



**:SOURce[1..n]:DATA:TELEcom:CFP:CSETting?**

<b>Description</b>	<p>This query returns the status of Control settings for TX &amp; RX IC RST, TX Disable, Module Low Power Mode, Module Reset, CFP Power Shutdown.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; CFP Control Pins</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:CFP:CSETting? &lt;wsp&gt;TXRXRST   TXDis   MOLOpwn   MODrst   CFPshutdown</p>
<b>Parameter(s)</b>	<p><b>State:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: TXRXRST   TXDis   MOLOpwn   MODrst   CFPshutdown</p> <p>Selects the Control settings.</p> <p>TXRXRST: TX &amp; RX IC RST.</p> <p>TXDis: TX Disable.</p> <p>MOLOpwn: Module Low Power Mode.</p> <p>MODrst: Module Reset.</p> <p>CFPShutdown: CFP Power Shutdown.</p>
<b>Response Syntax</b>	<p>&lt;Set&gt;</p>

## SCPI Command Reference

### 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

---

**:SOURce[1..n]:DATA:TELEcom:CFP:CSETting?**

<b>Response(s)</b>	<b>Set:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the state of selected control settings. TXRXRST, selects the TX & RX IC RST. TXDIs, selects the TX Disable. MOLopwn, selects Module Low Power Mode. MODRst, selects Module Reset. CFPShutdown, selects CFP Power Shutdown.
<b>Example(s)</b>	SOUR:DATA:TEL:CFP:CSET CFP,ON SOUR:DATA:TEL:CFP:CSET? CFP Returns: 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:CFP:STATus?

---

:SOURce[1..n]:DATA:TELEcom:CFP:CPRating

<b>Description</b>	<p>This command selects the Connector Power Ratings.</p> <p>At *RST condition, this value is set to 8W.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; Connector Power Rating</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CFP:CPRating <wsp>8W   16W   24W   32W   3W   6W   9W   12W
<b>Parameter(s)</b>	<p><b>Rating:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 8W   16W   24W   32W   3W   6W   9W   12W</p> <p>This command selects the Connector Power Rating.</p> <p>8W: 8W Connector Power Rating for CFP.</p> <p>16W: 16W Connector Power Rating for CFP.</p> <p>24W: 24W Connector Power Rating for CFP.</p> <p>32W: 32W Connector Power Rating for CFP.</p> <p>3W: 3W Connector Power Rating for CFP2.</p> <p>6W: 6W Connector Power Rating for CFP2.</p> <p>9W: 9W Connector Power Rating for CFP2.</p> <p>12W: 12W Connector Power Rating for CFP2.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CFP:CPR 8W</p> <p>SOUR:DATA:TEL:CFP:CPR?</p> <p>Returns: 8W</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:CFP:CSET</p> <p>SOURce[1..n]:DATA:TELEcom:CFP:CSET?</p>

## SCPI Command Reference

### 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

---

**:SOURce[1..n]:DATA:TELEcom:CFP:CPRating?**

<b>Description</b>	<p>This query returns the Connector Power Ratings.</p> <p>At *RST condition, this value is set to 8W.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; Connector Power Rating</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CFP:CPRating?
<b>Response Syntax</b>	<Rating>
<b>Response(s)</b>	<p><b>Rating:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>This query returns the Connector Power Rating.</p> <p>8W, selects 8W Connector Power Rating for CFP.</p> <p>16W, selects 16W Connector Power Rating for CFP.</p> <p>24W, selects 24W Connector Power Rating for CFP.</p> <p>32W, selects 32W Connector Power Rating for CFP.</p> <p>3W, selects 3W Connector Power Rating for CFP2.</p> <p>6W, selects 6W Connector Power Rating for CFP2.</p> <p>9W, selects 9W Connector Power Rating for CFP2.</p> <p>12W, selects 12W Connector Power Rating for CFP2.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:CFP:CPR?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:CFP:CSET SOURce[1..n]:DATA:TELEcom:CFP:CSET?

---

**:SOURce[1..n]:DATA:TELEcom:CFP:STATus?**

<b>Description</b>	<p>This query returns the status of the CFP status pins.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; CFP Status Pins</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:CFP:STATus? &lt;wsp&gt;HPDstate   NREady   MFAULT   MABSent   RXLos   AActive</p>
<b>Parameter(s)</b>	<p><b>State:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: HPDstate   NREady   MFAULT   MABSent   RXLos   AActive</p> <p>Returns the status of CFP Status Pins.</p> <p>HPDState: HI Power On.</p> <p>NREady: Module Ready.</p> <p>MFAULT: Module Fault.</p> <p>MABSent: Module Absent.</p> <p>RXLos: RX Loss of Signal.</p> <p>AActive: Global Alarm.</p>
<b>Response Syntax</b>	<p>&lt;Status&gt;</p>

## SCPI Command Reference

### 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

---

:SOURce[1..n]:DATA:TELEcom:CFP:STATus?

<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the status of the CFP Status Pins. UP, indicates the HI power ON. DOWN, indicates the HI power OFF. READY, indicates the Module is ready. NOTREADY, indicates the Module is not ready. NFAULT, indicates the Module is not faulty. FAULT, indicates the Module is faulty. PRESENT, indicates the CFP Status Pins is present. ABSENT, indicates the CFP Status Pins is absent. OK, indicates the signal is OK. RXLOS, indicates the loss of signal. ALARM, indicates the global alarm is present. NOALARM, indicates the global alarm is absent.
<b>Example(s)</b>	SOUR:DATA:TEL:CFP:STAT? HPD SOUR:DATA:TEL:CFP:STAT? NRE SOUR:DATA:TEL:CFP:STAT? MAB SOUR:DATA:TEL:CFP:STAT? RXL SOUR:DATA:TEL:CFP:STAT? AAC SOUR:DATA:TEL:CFP:STAT? MFA
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:CFP:CPRating SOURce[1..n]:DATA:TELEcom:CFP:CPRating?

---

---

**:SENSe[1..n]:DATA:TELEcom:CFP:TX:STATus?**

<b>Description</b>	<p>This query returns the CFP TX status.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; CFP TX Status</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:CFP:TX:STATus?
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for status of the CFP TX Pins.</p>
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the status of the CFP TX pins.</p> <p>INLOC, indicates the optical lane is in LOC.</p> <p>NOTINLOC, indicates the optical lane is not in LOC.</p>
<b>Example(s)</b>	SENS:DATA:TEL:CFP:TX:STAT? 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:CFP:STATus?

---

## SCPI Command Reference

### 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

---

**:SOURce[1..n]:DATA:TELeom:CFP:RCLock**

<b>Description</b>	<p>This command selects the reference clock value.</p> <p>At *RST condition, this value is set to setup-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; CFP Reference Clock</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELeom:CFP:RCLock &lt;wsp&gt; &lt;Clock&gt;</code>
<b>Parameter(s)</b>	<p><b>Clock:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: 168.0407   672.1627   174.7031   698.8123   161.1328   644.5313</p> <p>Sets the reference clock value.</p> <p>168.0407: 168.0407</p> <p>672.1627: 672.1627 for OTU3</p> <p>174.7031: 174.7031 for OTU4</p> <p>698.8123: 698.8123 for OTU4</p> <p>161.1328: 161.1328 for 40G and 100G</p> <p>644.5313: 644.5313 for 40G and 100G</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:CFP:RCL 161.1328</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELeom:CFP:RCLock?</code>



---

**:SOURce[1..n]:DATA:TELEcom:CFP:RCLock?**

---

<b>Description</b>	<p>This query returns the reference clock value.</p> <p>At *RST condition, this value is set to setup-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; CFP Reference Clock</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CFP:RCLock?
<b>Response Syntax</b>	<clockValue>
<b>Response(s)</b>	<p><b>clockValue:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>This query returns the reference clock value.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:CFP:RCL?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:CFP:RCLock

---

## SCPI Command Reference

### 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

---

**:SOURce[1..n]:DATA:TELEcom:MDIO:ADDRess**

<b>Description</b>	<p>This command sets the address for Management Data Input/Output (MDIO). At *RST condition, this value is set to 0000.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:MDIO:ADDRess &lt;wsp&gt; &lt;Address&gt;</code>
<b>Parameter(s)</b>	<p><b>Address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the address for Management Data Input/Output (MDIO).</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:MDIO:ADDR #H0005 SOUR:DATA:TEL:MDIO:ADDR? Returns: 5</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:MDIO:DATA SOURce[1..n]:DATA:TELEcom:MDIO:DATA?</pre>

---

---

**:SOURce[1..n]:DATA:TELEcom:MDIO:ADDRess?**

---

<b>Description</b>	<p>This query returns the address for Management Data Input/Output (MDIO).</p> <p>At *RST condition, this value is set to 0000.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:MDIO:ADDRess?
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the address for Management Data Input/Output (MDIO).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:MDIO:ADDR #H0005</p> <p>SOUR:DATA:TEL:MDIO:ADDR?</p> <p>Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:MDIO:DATA</p> <p>SOURce[1..n]:DATA:TELEcom:MDIO:DATA?</p>

---

## SCPI Command Reference

### 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

---

**:SOURce[1..n]:DATA:TELEcom:MDIO:DATA**

<b>Description</b>	<p>This command sets the data for Management Data Input/Output (MDIO).</p> <p>At *RST condition, this value is set to 0000.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:MDIO:DATA &lt;wsp&gt; &lt;Data&gt;</p>
<b>Parameter(s)</b>	<p><b>Data:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the data for Management Data Input/Output (MDIO).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:MDIO:DATA #H0005</p> <p>SOUR:DATA:TEL:MDIO:DATA?</p> <p>Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:MDIO:ADDRess</p> <p>SOURce[1..n]:DATA:TELEcom:MDIO:ADDRess?</p>

---

**:SOURce[1..n]:DATA:TELEcom:MDIO:DATA?**

<b>Description</b>	<p>This query returns the data for Management Data Input/Output (MDIO).</p> <p>At *RST condition, this value is set to 0000.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:MDIO:DATA?
<b>Response Syntax</b>	<Data>
<b>Response(s)</b>	<p><b>Data:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the data for Management Data Input/Output (MDIO).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:MDIO:DATA #H0005</p> <p>SOUR:DATA:TEL:MDIO:DATA?</p> <p>Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:MDIO:ADDRess</p> <p>SOURce[1..n]:DATA:TELEcom:MDIO:ADDRess?</p>

## SCPI Command Reference

### 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

---

**:SOURce[1..n]:DATA:TELEcom:MDIO:READ**

<b>Description</b>	<p>This command reads the Management Data Input/Output (MDIO) values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:MDIO:READ
<b>Example(s)</b>	SOUR:DATA:TEL:MDIO:READ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:MDIO:WRITE

---

**:SOURce[1..n]:DATA:TELEcom:MDIO:WRITe**

<b>Description</b>	<p>This command writes the Management Data Input/Output (MDIO) values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:MDIO:WRITe
<b>Example(s)</b>	SOUR:DATA:TEL:MDIO:WRIT
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:MDIO:READ

## SCPI Command Reference

### 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

---

**:SOURce[1..n]:DATA:TELEcom:MDIO:STARt:ADDRess**

<b>Description</b>	<p>This command sets the start address for Management Data Input/Output (MDIO). At *RST condition, this value is set to 0000.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:MDIO:STARt:ADDRess &lt;wsp&gt; &lt;StartAddress&gt;</code>
<b>Parameter(s)</b>	<p><b>StartAddress:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the start address for Management Data Input/Output (MDIO).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:MDIO:STAR:ADDR #H0005</p> <p>SOUR:DATA:TEL:MDIO:STAR:ADDR?</p> <p>Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:MDIO:END:ADDRess</p> <p>SOURce[1..n]:DATA:TELEcom:MDIO:END:ADDRess?</p>

---



**:SOURce[1..n]:DATA:TELEcom:MDIO:STARt:ADDRess?**

<b>Description</b>	<p>This query returns the start address for Management Data Input/Output (MDIO).                  At *RST condition, this value is set to 0000.                  Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:MDIO:STARt:ADDRess?
<b>Response Syntax</b>	<StartAddress>
<b>Response(s)</b>	<p><b>StartAddress:</b>                  The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.                  Returns the start address for Management Data Input/Output (MDIO).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:MDIO:STAR:ADDR #H0005                  SOUR:DATA:TEL:MDIO:STAR:ADDR?                  Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:MDIO:END:ADDRess                  SOURce[1..n]:DATA:TELEcom:MDIO:END:ADDRess?</p>

---

## SCPI Command Reference

### 40/100G Advanced - CFP/CFP2/CFP4/QSFP Control

---

**:SOURce[1..n]:DATA:TELEcom:MDIO:END:ADDRess**

<b>Description</b>	<p>This command sets the end address for Management Data Input/Output (MDIO).</p> <p>At *RST condition, this value is set to 0000.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:MDIO:END:ADDRess &lt;wsp&gt; &lt;EndAddress&gt;</p>
<b>Parameter(s)</b>	<p><b>EndAddress:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the end address for Management Data Input/Output (MDIO).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:MDIO:END:ADDR #H0005</p> <p>SOUR:DATA:TEL:MDIO:END:ADDR?</p> <p>Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:MDIO:STARt:ADDRess</p> <p>SOURce[1..n]:DATA:TELEcom:MDIO:STARt:ADDRess?</p>

---

**:SOURce[1..n]:DATA:TELEcom:MDIO:END:ADDRess?**

<b>Description</b>	<p>This query returns the end address for Management Data Input/Output (MDIO).                  At *RST condition, this value is set to 0000.                  Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:MDIO:END:ADDRess?
<b>Response Syntax</b>	<EndAddress>
<b>Response(s)</b>	<p><b>EndAddress:</b>                  The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.                  Returns the end address for Management Data Input/Output (MDIO).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:MDIO:END:ADDR #H0005                  SOUR:DATA:TEL:MDIO:END:ADDR?                  Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:MDIO:STARt:ADDRess                  SOURce[1..n]:DATA:TELEcom:MDIO:STARt:ADDRess?</p>

---

## 40/100G Advanced - Pre-Emphasis

---

:SOURce[1..n]:DATA:TELEcom:CONFig

---

<b>Description</b>	<p>This command enables and disables All channel configuration.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; All Lanes</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CONFig <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables and disables the All channel configuration.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CONF OFF</p> <p>SOUR:DATA:TEL:CONF?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TRAN:RX:ECOntrl

---

---

**:SOURce[1..n]:DATA:TELEcom:CONFig?**

---

<b>Description</b>	This query returns the status of an All Channel Configuration. At *RST condition, this value is set to OFF. Navigation Path: Functions > 40/100G Advanced > Pre-Emphasis > All Lanes
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CONFig?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<b>Status:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of all the channels configuration. 1, returns All channel configuration as ON. 0, returns All channel configuration as OFF.
<b>Example(s)</b>	SOUR:DATA:TEL:CONF ON SOUR:DATA:TEL:CONF? Returns: 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TRAN:RX:ECOntrOl?

---

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

:SOURce[1..n]:DATA:TELEcom:TRANsceiver:VOD

<b>Description</b>	<p>This command sets the channel signal amplitude.</p> <p>At *RST condition, this value is set to 800.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx &gt; VOD (mV)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TRANsceiver:VOD <wsp> <Channel>, 800   200   400   600   700   900   1000   1200
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Allows to change the channel's signal amplitude.</p> <p><b>Vod:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 800   200   400   600   700   900   1000   1200</p> <p>Sets the channel's signal amplitude.</p> <p>800, sets the channel's signal amplitude to 800.</p> <p>200, sets the channel's signal amplitude to 200.</p> <p>400, sets the channel's signal amplitude to 400.</p> <p>600, sets the channel's signal amplitude to 600.</p> <p>700, sets the channel's signal amplitude to 700.</p> <p>900, sets the channel's signal amplitude to 900.</p> <p>1000, sets the channel's signal amplitude to 1000.</p> <p>1200, sets the channel's signal amplitude to 1200.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:VOD 1, 800</p> <p>SOUR:DATA:TEL:TRAN:VOD? 1</p> <p>Returns: 800</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:VOD</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:VOD?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap?</p>

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:VOD?**

---

<b>Description</b>	<p>The query returns the channel signal amplitude.</p> <p>At *RST condition, this value is set to 800.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx &gt; VOD (mV)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:VOD? &lt;wsp&gt; &lt;Channel&gt;</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Returns the channel signal amplitude.</p>
<b>Response Syntax</b>	<p>&lt;Vod&gt;</p>
<b>Response(s)</b>	<p><b>Vod:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the channel's signal amplitude.</p> <p>800, sets the channel's signal amplitude to 800.</p> <p>200, sets the channel's signal amplitude to 200.</p> <p>400, sets the channel's signal amplitude to 400.</p> <p>600, sets the channel's signal amplitude to 600.</p> <p>700, sets the channel's signal amplitude to 700.</p> <p>900, sets the channel's signal amplitude to 900.</p> <p>1000, sets the channel's signal amplitude to 1000.</p> <p>1200, sets the channel's signal amplitude to 1200.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:VOD? 1</p> <p>SOUR:DATA:TEL:TRAN:VOD 1, 800</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:VOD</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:VOD?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap?</p>

---

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:PRET**

<b>Description</b>	<p>This command sets the Pre-tap signal for Pre-Emphasis characteristic of the signal in TX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:PRET &lt;wsp&gt; &lt;Channel&gt;, &lt;Types&gt;</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel for setting Pre-tap signal for Pre-Emphasis characteristic of the signal in TX</p> <p><b>Types:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Transmit Transceiver signal level for TX.</p> <p>-15 to +15 sets the signal in TX.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:TX:PEMP:PRET 1, 15</p> <p>SOUR:DATA:TEL:TRAN:TX:PEMP:PRET? 1</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:POSTtap1?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:POSTtap1</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:PRET?**

---

<b>Description</b>	<p>This query returns the Pre-tap signal for Pre-Emphasis characteristic of the signal in TX. At *RST condition, this value is set to device-dependent. Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:PRET? &lt;wsp&gt; &lt;Channel&gt;</p>
<b>Parameter(s)</b>	<p><b>Channel:</b> The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element. Selects the channel for getting Pre-tap signal for Pre-Emphasis characteristic of the signal in TX</p>
<b>Response Syntax</b>	<p>&lt;Type&gt;</p>
<b>Response(s)</b>	<p><b>Type:</b> The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element. Returns the selected signal in TX.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:TX:PEMP:PRET? 1 SOUR:DATA:TEL:TRAN:TX:PEMP:PRET 1, 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap? SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:POSTtap1? SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:POSTtap1</p>

---

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:POST[1..n]**

<b>Description</b>	<p>This command sets the Post-tap signal for Pre-Emphasis characteristic of the signal in TX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:POST[1..n] &lt;wsp&gt; &lt;Channel&gt;, &lt;Types&gt;</pre>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel for setting Post-tap signal for Pre-Emphasis characteristic of the signal in TX.</p> <p><b>Types:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the transmit transceiver signal level for TX.</p> <p>""0"" to ""31"" (Pre Emphasis Post Tap 1t), ""0"" to ""31"" sets the signal in TX</p> <p>""-15"" to ""+15"" (Pre Emphasis Post Tap 2t), ""-15"" to ""+15"" sets the signal in TX.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TRAN:TX:PEMP:POST1? 1 SOUR:DATA:TEL:TRAN:TX:PEMP:POST1 1, 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:POSTtap1? SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:POSTtap1 SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap? SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap</pre>

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:POST[1..n]?**

<b>Description</b>	<p>This query returns the Post-tap signal for Pre-Emphasis characteristic of the signal in TX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:POST[1..n]? <wsp> <Channel>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel for getting Post-tap signal for Pre-Emphasis characteristic of the signal in TX.</p>
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Post-tap signal's for Pre-Emphasis characteristic of the signal in TX.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:TX:PEMP:POST1 1, 15</p> <p>SOUR:DATA:TEL:TRAN:TX:PEMP:POST1? 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:POSTtap1</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:POSTtap1?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:PRETap</p>

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APR**

<b>Description</b>	<p>This command sets the Pre-tap signal for Pre-Emphasis characteristic of the signal in TX for all channels.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APR &lt;wsp&gt; &lt;Types&gt;</code>
<b>Parameter(s)</b>	<p><b>Types:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Pre-tap signal's for Pre-Emphasis characteristic of the signal in TX for all channel.</p> <p>-15 to +15 sets the signal in TX.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TRAN:TX:PEMP:APR 15 SOUR:DATA:TEL:TRAN:TX:PEMP:APR? Returns: 15</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap? SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APOSTtap1? SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APOSTtap1</pre>

---

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APR?**

---

<b>Description</b>	<p>This query returns the Pre-tap signal for Pre-Emphasis characteristic of the signal in TX for all channels.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APR?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Pre-tap signal's for Pre-Emphasis characteristic of the signal in TX for all channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:TX:PEMP:APR 15</p> <p>SOUR:DATA:TEL:TRAN:TX:PEMP:APR?</p> <p>Returns: 15</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APOSTtap1?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APOSTtap1</p>

---

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APOS[1..n]**

<b>Description</b>	<p>This command sets the Post-tap signal for Pre-Emphasis characteristic of the signal in TX for all channels.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APOS[1..n] &lt;wsp&gt; &lt;Types&gt;</code> ,
<b>Parameter(s)</b>	<p><b>Types:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Post-tap signal's for Pre-Emphasis characteristic of the signal in TX for all channel.</p> <p>""0"" to ""31"" (Pre Emphasis Post Tap 1t), ""0"" to ""31"" sets the signal in TX.</p> <p>""-15"" to ""+15"" (Pre Emphasis Post Tap 2t), ""-15"" to ""+15"" sets the signal in TX.</p> <p>The allowed elements for this parameter are:</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TRAN:TX:PEMP:APOS1 1 SOUR:DATA:TEL:TRAN:TX:PEMP:APOS1? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APOSTtap1? SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APOSTtap1 SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap? SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap</pre>

---

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APOS[1..n]?**

---

<b>Description</b>	<p>This query returns the Post-tap signal for Pre-Emphasis characteristic of the signal in TX for all channel.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TRANsceiver:TX:PEMP:APOS[1..n]?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Post-tap signal's for Pre-Emphasis characteristic of the signal in TX for all channel.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:TX:PEMP:APOS1 1</p> <p>SOUR:DATA:TEL:TRAN:TX:PEMP:APOS1?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APOSTtap1</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APOSTtap1?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap</p>

---

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:ECONtrol**

<b>Description</b>	<p>This command sets the Equalizer Control signals to RX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Rx</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:ECONtrol &lt;wsp&gt; &lt;Channel&gt;, ""0"" to ""15""</pre>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel for setting Equalizer Control signals to RX.</p> <p><b>Types:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: ""0"" to ""15""</p> <p>Selects the Transmit Transceiver signal level for RX.</p> <p>""0"" to ""15"", 0 to 15 sets the signal in RX.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TRAN:RX:ECON 1, 5 SOUR:DATA:TEL:TRAN:RX:ECON? 1 Returns: 5</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:TRAN:RX:ECONtrol SOURce[1..n]:DATA:TELEcom:TRAN:RX:ECONtrol? SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap? SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap</pre>

---



---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:ECONtrol?**

---

<b>Description</b>	<p>This query returns the Equalizer Control signals to RX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Rx</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:ECONtrol? &lt;wsp&gt; &lt;Channel&gt;,</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel for getting Equalizer Control signals to RX.</p> <p>The allowed elements for this parameter are:</p>
<b>Response Syntax</b>	<p>&lt;Type&gt;</p>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Equalizer Control signals to RX.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:RX:ECON 1, 5</p> <p>SOUR:DATA:TEL:TRAN:RX:ECON? 1</p> <p>Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:ECONtrol?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:ECONtrol</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:TX:PEMPhasis:APRetap</p>

---

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:EGain**

<b>Description</b>	<p>This command sets the Equalizer Gain signals for RX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Rx</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:EGain &lt;wsp&gt; &lt;Channel&gt;, 0   3   6   9   12</pre>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel for setting Equalizer Gain signals to RX.</p> <p><b>Types:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 0   3   6   9   12</p> <p>Selects the transmit transceiver signal level for RX.</p> <p>0: 0 as Equalizer Gain for RX</p> <p>3: 3 as Equalizer Gain for RX</p> <p>6: 6 as Equalizer Gain for RX</p> <p>9: 9 as Equalizer Gain for RX</p> <p>12: 12 as Equalizer Gain for RX</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TRAN:RX:EGA 1, 3 SOUR:DATA:TEL:TRAN:RX:EGA? 1 Returns: 3</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:TRAN:RX:EGain SOURce[1..n]:DATA:TELEcom:TRAN:RX:EGain? SOURce[1..n]:DATA:TELEcom:TRAN:RX:ECOntrol? SOURce[1..n]:DATA:TELEcom:TRAN:RX:ECOntrol</pre>

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:EGain?**

---

<b>Description</b>	<p>This query returns the Equalizer Gain signals to RX.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Rx</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:EGain? &lt;wsp&gt; &lt;Channel&gt;</p>
<b>Parameter(s)</b>	<p><b>Channel:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel for getting Equalizer Gain signals to RX.</p>
<b>Response Syntax</b>	<p>&lt;Type&gt;</p>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Equalizer Gain signals for RX.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:RX:EGA? 1</p> <p>SOUR:DATA:TEL:TRAN:RX:EGA 1, 3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:EGain?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:EGain</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:ECOntrol?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:ECOntrol</p>

---

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:AVOD**

<b>Description</b>	<p>This command sets the channel's signal amplitude for all channels.</p> <p>At *RST condition, this value is set to 800.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx &gt; AVOD (mV)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:AVOD &lt;wsp&gt;800   200   400   600   700   900   1000   1200</p>
<b>Parameter(s)</b>	<p><b>AVod:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 800   200   400   600   700   900   1000   1200</p> <p>Sets the channel's signal amplitude.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:AVOD 800</p> <p>SOUR:DATA:TEL:TRAN:AVOD?</p> <p>Returns: 800</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:AVOD</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:AVOD?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:AECControl</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:AECControl?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:AVOD?**

---

<b>Description</b>	<p>This query returns the channel's signal amplitude for all channels. At *RST condition, this value is set to 800. Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Tx &gt; AVOD (mV)</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:AVOD?</p>
<b>Response Syntax</b>	<p>&lt;Avod&gt;</p>
<b>Response(s)</b>	<p><b>Avod:</b> The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element. Returns the channel's signal amplitude: 800, 200, 400, 600, 700, 900, 1000, or 1200.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:AVOD 800 SOUR:DATA:TEL:TRAN:AVOD? Returns: 800</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:AVOD SOURce[1..n]:DATA:TELEcom:TRAN:AVOD? SOURce[1..n]:DATA:TELEcom:TRAN:RX:AECControl SOURce[1..n]:DATA:TELEcom:TRAN:RX:AECControl?</p>

---

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

---

#### :SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:AECControl

---

<b>Description</b>	<p>This command sets the Equalizer Control signals for RX for all channels.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Rx</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:AECControl <wsp>""0"" to ""15""
<b>Parameter(s)</b>	<p><b>Types:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: ""0"" to ""15""</p> <p>Selects the Transmit Transceiver signal level for RX.</p> <p>""0"" to ""15"", 0 to 15 sets the signal in RX.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:RX:AEC 5</p> <p>SOUR:DATA:TEL:TRAN:RX:AEC?</p> <p>Returns: 5</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:AECControl</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:AECControl?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:AVOD?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:AVOD</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:AEControl?**

---

<b>Description</b>	This query returns the Equalizer Control signals to RX for all channels. At *RST condition, this value is set to device-dependent. Navigation Path: Functions > 40/100G Advanced > Pre-Emphasis > Rx
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:AEControl?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<b>Type:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the selected equalizer control signal in RX.
<b>Example(s)</b>	SOUR:DATA:TEL:TRAN:RX:AEC 5 SOUR:DATA:TEL:TRAN:RX:AEC? Returns: 5
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TRAN:RX:AEControl SOURce[1..n]:DATA:TELEcom:TRAN:RX:AEControl? SOURce[1..n]:DATA:TELEcom:TRAN:AVOD? SOURce[1..n]:DATA:TELEcom:TRAN:AVOD

---

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:AEGain**

<b>Description</b>	<p>This command sets the Equalizer Gain signals for RX for all channels.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Rx</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:AEGain &lt;wsp&gt;0   3   6   9   12</code>
<b>Parameter(s)</b>	<p><b>Types:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: 0   3   6   9   12</p> <p>Selects the transmit transceiver signal level for RX.</p> <p>0: 0 as Equalizer Gain for RX</p> <p>3: 3 as Equalizer Gain for RX</p> <p>6: 6 as Equalizer Gain for RX</p> <p>9: 9 as Equalizer Gain for RX</p> <p>12: 12 as Equalizer Gain for RX</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TRAN:RX:AEG 3 SOUR:DATA:TEL:TRAN:RX:AEG? Returns: 3</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:TRAN:RX:AEGain SOURce[1..n]:DATA:TELEcom:TRAN:RX:AEGain? SOURce[1..n]:DATA:TELEcom:TRAN:AVOD? SOURce[1..n]:DATA:TELEcom:TRAN:AVOD</pre>

---



---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:AEGain?**

---

<b>Description</b>	<p>This query returns the Equalizer Gain signals to RX for all channels.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Rx</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RX:AEGain?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the selected equalizer gain signal in RX.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAN:RX:AEG 3</p> <p>SOUR:DATA:TEL:TRAN:RX:AEG?</p> <p>Returns: 3</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:AEGain?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:RX:AEGain</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:AVOD?</p> <p>SOURce[1..n]:DATA:TELEcom:TRAN:AVOD</p>

---

## SCPI Command Reference

### 40/100G Advanced - Pre-Emphasis

---

**:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RESet**

<b>Description</b>	<p>This command resets all the pre-emphasis parameters to their default values.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Functions &gt; 40/100G Advanced &gt; Pre-Emphasis &gt; Reset</p>
<b>Syntax</b>	<pre>:SOURce[1..n]:DATA:TELEcom:TRANsceiver:RESet</pre>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TRAN:RES SOUR:DATA:TEL:TRAN:RX:AEG? Returns: 3</pre>

---

## Default/Random/Manual Mapping

:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:TX

<b>Description</b>	<p>This command sets the TX Lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Mapping</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:TX &lt;wsp&gt; &lt;LaneNo&gt;, PCS   SKEW, &lt;Value&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>LaneNo:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the selected layer.</p> <p>The Lane number used for the skew is a logical lane number. The Lane number used for PCS is a physical lane number.</p> <p><b>Layer Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCS   SKEW</p> <p>Selects the Link Layer Type.</p> <p>SKEW: SKEW</p> <p>PCS: PCS</p> <p><b>Value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:LLAY:TX 5,PCS,2</p> <p>SOUR:DATA:TEL:ETH:LLAY:TX? 5,PCS</p> <p>Returns: 2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX</p>

## SCPI Command Reference

### Default/Random/Manual Mapping

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:TX?**

<b>Description</b>	<p>This query returns the TX Lane.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Mapping</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:TX? &lt;wsp&gt; &lt;LaneNo&gt;, PCS   SKEW[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>LaneNo:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the lane for the layer.</p> <p>The Lane number used for the skew is a logical lane number. The Lane number used for PCS is a physical lane number.</p> <p><b>Layer Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCS   SKEW</p> <p>Selects the Link Layer Type.</p> <p>SKEW: SKEW</p> <p>PCS: PCS</p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p>
<b>Response Syntax</b>	<p>&lt;Llayer&gt;</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:TX?**

<b>Response(s)</b>	<p><b>Llayer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the link layer virtual to physical lane skew for the transmitter.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:LLAY:TX 5,PCS,2</p> <p>SOUR:DATA:TEL:ETH:LLAY:TX? 5,PCS</p> <p>Returns: 2</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX</p>

## SCPI Command Reference

### Default/Random/Manual Mapping

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:LLAYer:RX?**

<b>Description</b>	<p>This query returns the RX Lane.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:LLAYer:RX? &lt;wsp&gt; &lt;Physical Lane&gt;, PCS   SKEW</p>
<b>Parameter(s)</b>	<p><b>Physical Lane:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the physical lane skew. The range for the physical lane is from 0 to 19.</p> <p>The Lane number used for the skew is a logical lane number. The Lane number used for PCS is a physical lane number.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: PCS   SKEW</p> <p>Selects the Link Layer Type.</p> <p>SKEW: SKEW</p> <p>PCS: PCS</p>
<b>Response Syntax</b>	<p>&lt;Llayer&gt;</p>
<b>Response(s)</b>	<p><b>Llayer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the link layer virtual to physical lane skew for the receiver.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:LLAY:RX? 8,PCS</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:WIS:TRAcE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane**

---

<b>Description</b>	<p>This command sets the applied change(s) to all PCS/Logical Lane at once.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX &gt; All Lanes</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Sets the All lanes status.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:LLAY:SKEW:ALLL OFF</p> <p>SOUR:DATA:TEL:ETH:LLAY:SKEW:ALLL?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETH:LLAY:SKEW:ALLL:TX 50

---

## SCPI Command Reference

### *Default/Random/Manual Mapping*

---

**:SOURce[1..n]:DATA:TELeom:ETHernet:LLAYer:SKEW:ALLLane?**

<b>Description</b>	<p>This query returns the status of the applied change(s) to all PCS/Logical Lane.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX &gt; All Lanes</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELeom:ETHernet:LLAYer:SKEW:ALLLane?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the All lanes status.</p> <p>1, All lane status enabled.</p> <p>0, All lane status disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:LLAY:SKEW:ALLL ON</p> <p>SOUR:DATA:TEL:ETH:LLAY:SKEW:ALLL?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELeom:ETHernet:LLAYer:SKEW:ALLLane:TX

---



---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX**

---

<b>Description</b>	<p>This command sets the value for all PCS/Logical Lane when the All Lanes check box is selected.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX &gt; All</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX <wsp> <SKEW>
<b>Parameter(s)</b>	<p><b>SKEW:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the skew for all.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:LLAY:SKEW:ALLL:TX 50</p> <p>SOUR:DATA:TEL:ETH:LLAY:SKEW:ALLL:TX?</p> <p>Returns: 50</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane

---

## SCPI Command Reference

### *Default/Random/Manual Mapping*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX?**

<b>Description</b>	<p>This query returns the value for all PCS/Logical Lane when the All Lanes check box is selected.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX &gt; All</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane:TX?
<b>Response Syntax</b>	<Skew>
<b>Response(s)</b>	<p><b>Skew:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the all lane Skew value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:LLAY:SKEW:ALLL:TX 50</p> <p>SOUR:DATA:TEL:ETH:LLAY:SKEW:ALLL:TX?</p> <p>Returns: 50</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:LLAYer:SKEW:ALLLane?

---

## Reset/Manual Skew

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane**

---

<b>Description</b>	<p>This command sets the All Lane status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: OTN Bert Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX &gt; All Lanes</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Sets the All lanes status.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:SKEW:ALL 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX?

---

## SCPI Command Reference

### *Reset/Manual Skew*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane?**

<b>Description</b>	<p>This query returns the All Lane status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: OTN Bert Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX &gt; All Lanes</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the All lanes status.</p> <p>1, returns the all lane status as ON.</p> <p>0, returns the all lane status as OFF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTL:SKEW:ALLL ON</p> <p>SOUR:DATA:TEL:OTN:OTL:SKEW:ALLL?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX?</p>

---

---

**:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX**

---

<b>Description</b>	<p>This command sets the skew value for all lanes for the transmitted link.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: OTN Bert Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX &gt; All &gt; Skew Inc/Dec Size</p>
<b>Syntax</b>	<code>:SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX &lt;wsp&gt; &lt;SKEW&gt;</code>
<b>Parameter(s)</b>	<p><b>SKEW:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the skew for all.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:OTN:OTL:SKEW:ALLL:TX 50</code>
<b>See Also</b>	<code>SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane</code> <code>SOURCE[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane?</code>

---

## SCPI Command Reference

### *Reset/Manual Skew*

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX?**

<b>Description</b>	<p>This query returns the All Lane status for transmitted link.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: OTN Bert Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX &gt; All</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane:TX?
<b>Response Syntax</b>	<SKEW>
<b>Response(s)</b>	<p><b>SKEW:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the all lane Skew value.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:OTN:OTL:SKEW:ALL:TX?
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:OTL:SKEW:ALLLane?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:TX**

<b>Description</b>	<p>This command sets the value of the transmitted bits which depends on the selected link layer type.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: OTN BERT Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:TX &lt;wsp&gt; &lt;LaneNo&gt;, LOGicallane   SKEW, &lt;Value&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>LaneNo:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The Lane used for the skew is a logical lane number.</p> <p>The Lane used for LOG is a physical lane number.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOGicallane   SKEW</p> <p>Selects the Link Layer Type.</p> <p>SKEW: SKEW</p> <p>LOGicallane: Logical lane</p> <p><b>Value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTL:TX 3,SKEW,MIN</p> <p>SOUR:DATA:TEL:OTN:OTL:TX 5,LOG,2</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:OTN:OTL:RX?</p>

---

## SCPI Command Reference

### Reset/Manual Skew

---

**:SOURce[1..n]:DATA:TELEcom:OTN:OTL:TX?**

<b>Description</b>	<p>This query returns the value of the transmitted bits which depends on the selected link layer.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: OTN BERT Test &gt; Functions &gt; 40/100G Advanced &gt; Lanes Mapping &amp; Skew &gt; Manual Skew TX</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTL:TX? &lt;wsp&gt; &lt;LaneNo&gt;, LOGicallane   SKEW[, &lt;Value&gt;   MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>LaneNo:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The Lane used for the skew is a logical lane number.</p> <p>The Lane used for LOG is a physical lane number.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: LOGicallane   SKEW</p> <p>Selects the Link Layer Type.</p> <p>SKEW: SKEW</p> <p>LOGicallane: Logical lane</p> <p><b>Value:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p>
<b>Response Syntax</b>	<p>&lt;Llayer&gt;</p>
<b>Response(s)</b>	<p><b>Llayer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the link layer virtual to physical lane skew for the transmitter.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:OTN:OTL:TX? 5,LOG</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:OTN:OTL:TX?</p>



## Bulk Read

---

**:SOURce[1..n]:DATA:TELEcom:MDIO:BULK:READ**

---

<b>Description</b>	<p>This command bulk reads the Management Data Input/Output (MDIO) values.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:MDIO:BULK:READ
<b>Example(s)</b>	SOUR:DATA:TEL:MDIO:BULK:READ
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:MDIO:READ

---

## SCPI Command Reference

### *Bulk Read*

---

---

#### :FETCh[1..n]:DATA:TELEcom:MDIO:BULK:READ:INFormation?

---

<b>Description</b>	<p>This query bulk reads the Management Data Input/Output (MDIO) values from given MDIO address range.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; 40/100G Advanced &gt; CFP Control &gt; MDIO Access Interface &gt; Bulk Read &gt; MDIO Access</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:MDIO:BULK:READ:INFormation?
<b>Response Syntax</b>	<Address and Data>
<b>Response(s)</b>	<p><b>Address and Data:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the list of address and data values.</p>
<b>Example(s)</b>	FETC:DATA:TEL:MDIO:BULK:READ:INF?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:MDIO:BULK:READ

---

## Ping & Trace Route

---

**:FETCh[1..n]:DATA:TELEcom:PING:STATistics:TX?**

---

<b>Description</b>	<p>This query returns the number of packets sent.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Results &gt; Statistics</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PING:STATistics:TX?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of packets sent.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PING:STAT:TX?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PING:STATistics:RX?

---

## SCPI Command Reference

### *Ping & Trace Route*

---

---

#### :FETCh[1..n]:DATA:TELEcom:PING:STATistics:RX?

---

<b>Description</b>	This query returns the number of packets received. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Functions > Ping & Trace Route > Results > Statistics
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PING:STATistics:RX?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<b>Result:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of packets received.
<b>Example(s)</b>	FETC:DATA:TEL:PING:STAT:RX?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PING:STATistics:LOST?

---

---

**:FETCh[1..n]:DATA:TELEcom:PING:STATistics:LOST?**

---

<b>Description</b>	<p>This query returns the percentage of lost packets.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Results &gt; Statistics</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PING:STATistics:LOST?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the percentage of lost packets.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PING:STAT:LOST?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PING:STATistics:MINimum?

---

## SCPI Command Reference

### *Ping & Trace Route*

---

---

#### :FETCh[1..n]:DATA:TELeom:PING:STATistics:MINimum?

---

<b>Description</b>	This query returns the minimum measured round trip time required for a Ping request. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Functions > Ping & Trace Route > Results > Statistics
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:PING:STATistics:MINimum?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<b>Result:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the minimum measured round trip time required for a Ping request.
<b>Example(s)</b>	FETC:DATA:TEL:PING:STAT:MIN?
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:PING:STATistics:MAXimum?

---

---

**:FETCh[1..n]:DATA:TELEcom:PING:STATistics:MAXimum?**

---

<b>Description</b>	This query returns the maximum measured round trip time required for a Ping request. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Functions > Ping & Trace Route > Results > Statistics
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PING:STATistics:MAXimum?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<b>Result:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the maximum measured round trip time required for a Ping request.
<b>Example(s)</b>	FETC:DATA:TEL:PING:STAT:MAX?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PING:STATistics:AVERage?

---

## SCPI Command Reference

### *Ping & Trace Route*

---

---

#### :FETCh[1..n]:DATA:TELEcom:PING:STATistics:AVERAge?

---

<b>Description</b>	This query returns the average measured round trip time required for a Ping request. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Functions > Ping & Trace Route > Results > Statistics
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PING:STATistics:AVERAge?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<b>Result:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the average measured round trip time required for a Ping request.
<b>Example(s)</b>	FETC:DATA:TEL:PING:STAT:AVER?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PING:STATistics:TX?

---



---

**:FETCh[1..n]:DATA:TELEcom:PING:STATistics:RESults?**

---

<b>Description</b>	<p>This query returns the list of Ping Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Results</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:PING:STATistics:RESults?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>This query returns the list of Ping Statistics.</p>
<b>Example(s)</b>	FETC:DATA:TEL:PING:STAT:RES?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:PING:STATistics:TX?

---

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:PING:SETup:RUN**

<b>Description</b>	<p>This command allows you to run the Ping command.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:SETup:RUN <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>SET:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the Ping command status.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:SET:RUN ON</p> <p>SOUR:DATA:TEL:PING:SET:RUN?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOUT 10000

---

---

**:SOURce[1..n]:DATA:TELEcom:PING:SETup:RUN?**

---

<b>Description</b>	<p>This query returns the status of the Ping command.</p> <p>This query is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:SETup:RUN?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Ping command.</p> <p>0, returns the status of the Ping command as OFF.</p> <p>1, returns the status of the Ping command as ON.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:SET:RUN ON</p> <p>SOUR:DATA:TEL:PING:SET:RUN?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOUT?

---

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOUT**

<b>Description</b>	<p>This command allows you to enter the maximum time allowed between an ICMP echo and response. Choices are 200 ms to 10000 ms.</p> <p>At *RST condition, this value is set to 4000 ms.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOUT &lt;wsp&gt; &lt;TOUT&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>TOUT:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the ping time out value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:TOUT 10000</p> <p>SOUR:DATA:TEL:PING:CONF:TOUT?</p> <p>Returns: 10000</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:DELay</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:DELay?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOUT?**

<b>Description</b>	<p>This query returns the time allowed between an ICMP echo and response. Choices are 200 ms to 10000 ms.</p> <p>At *RST condition, this value is set to 4000 ms.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOUT?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current time allowed between an ICMP echo and response is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the time allowed between an ICMP echo and response.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:TOUT 10000</p> <p>SOUR:DATA:TEL:PING:CONF:TOUT?</p> <p>Returns: 10000</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:DELAy</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:DELAy?</p>

---

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:DELay**

<b>Description</b>	<p>This command allows you to enter the delay between each attempt (PING). Choices are 100 to 10000 ms.</p> <p>At *RST condition, this value is set to 1000 ms.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:PING:CONFig:DELay &lt;wsp&gt; &lt;Delay&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Delay:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the ping delay value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:DEL 2000</p> <p>SOUR:DATA:TEL:PING:CONF:DEL?</p> <p>Returns: 2000</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:DSIZE</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:DSIZE?</p>

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:DELay?**

<b>Description</b>	<p>This query returns the delay between each attempt (PING). Choices are 100 to 10000 ms. At *RST condition, this value is set to 1000 ms. Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFig:DELay?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element. The allowed elements for this parameter are: MAXimum   MINimum This parameter is optional. If no token is specified, the current delay between each attempt (PING) is returned. MAXimum: Biggest supported value MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b> The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element. Returns the delay value between each attempt in the PING request.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:DEL 2000 SOUR:DATA:TEL:PING:CONF:DEL? Returns: 2000</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:DSIZe SOURce[1..n]:DATA:TELEcom:PING:CONFig:DSIZe?</p>

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:DSIZe**

<b>Description</b>	<p>This command allows you to enter the data size. Choices are 0 to 1472 Bytes.</p> <p>At *RST condition, this value is set to 32 bytes.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:PING:CONFig:DSIZe &lt;wsp&gt; &lt;Dsize&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Dsize:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the ping frame size value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:DSIZ 1460</p> <p>SOUR:DATA:TEL:PING:CONF:DSIZ?</p> <p>Returns: 1460</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TTL</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TTL?</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:DSIZe?**

<b>Description</b>	<p>This query returns the data size. Choices are 0 to 1472 Bytes.</p> <p>At *RST condition, this value is set to 32 bytes.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFig:DSIZe? [ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current data size is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the data size value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:DSIZ 1460</p> <p>SOUR:DATA:TEL:PING:CONF:DSIZ?</p> <p>Returns: 1460</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TTL</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TTL?</p>

---

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TTL**

<b>Description</b>	<p>This Command sets number of hops the packet can go through.</p> <p>At *RST condition, this value is set to 128.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TTL &lt;wsp&gt; &lt;Ttl&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Ttl:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the ping ttl value.</p> <p>Choices are 1 to 255.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:TTL 255</p> <p>SOUR:DATA:TEL:PING:CONF:TTL?</p> <p>Returns: 255</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOS</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOS?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TTL?**

---

<b>Description</b>	<p>This query returns the maximum number of hops the packet can go through.</p> <p>At *RST condition, this value is set to 128.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TTL?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current number of hops the packet can go through is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of hops the packet can go through.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:TTL 255</p> <p>SOUR:DATA:TEL:PING:CONF:TTL?</p> <p>Returns: 255</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOS</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOS?</p>

---

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOS**

<b>Description</b>	<p>This command allows you to enter the type of service.</p> <p>At *RST condition, this value is set to 0X00.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOS &lt;wsp&gt; &lt;TOS&gt;</code>
<b>Parameter(s)</b>	<p><b>TOS:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Ping TOS/DS value.</p> <p>Choices are 00 to FF.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:PING:CONF:TOS #H00 SOUR:DATA:TEL:PING:CONF:TOS? Returns: 0</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:PING:CONFig:HCOunt</code>

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOS?**

<b>Description</b>	<p>This query returns the type of service.</p> <p>At *RST condition, this value is set to 0X00.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOS?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the type of service value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:TOS #H00</p> <p>SOUR:DATA:TEL:PING:CONF:TOS?</p> <p>Returns: 0</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PING:CONFig:HCOunt

---

## SCPI Command Reference

### *Ping & Trace Route*

---

**:FETCh[1..n]:DATA:TELeom:TRACe:STATistics:TX?**

<b>Description</b>	This query returns the number of packets sent. At *RST condition, this value is set to device-dependent. Navigation Path: Test > Functions > Ping & Trace Route > Results > Statistics
<b>Syntax</b>	:FETCh[1..n]:DATA:TELeom:TRACe:STATistics:TX?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<b>Result:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the number of packets sent.
<b>Example(s)</b>	FETC:DATA:TEL:TRAC:STAT:TX?
<b>See Also</b>	FETCh[1..n]:DATA:TELeom:TRACe:STATistics:RX?

---

---

**:FETCh[1..n]:DATA:TELecom:TRACe:STATistics:RX?**

---

<b>Description</b>	<p>This query returns the number of packets received.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Results &gt; Statistics</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELecom:TRACe:STATistics:RX?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of packets received.</p>
<b>Example(s)</b>	FETC:DATA:TEL:TRAC:STAT:RX?
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:TRACe:STATistics:TX?

---

## SCPI Command Reference

### *Ping & Trace Route*

---

---

#### **:FETCh[1..n]:DATA:TELEcom:TRACe:STATistics:RESults?**

---

<b>Description</b>	<p>This query returns the list of Trace Statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Results</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:TRACe:STATistics:RESults?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>This query returns the list of Trace Statistics.</p>
<b>Example(s)</b>	FETC:DATA:TEL:TRAC:STAT:RES?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:TRACe:STATistics:TX?

---



**:SOURce[1..n]:DATA:TELEcom:TRACe:CONFIg:RUN**

<b>Description</b>	<p>This command allows you to run the Trace command.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Trace Route</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TRACe:CONFIg:RUN <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>SET:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the Trace command status.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAC:CONF:RUN ON</p> <p>SOUR:DATA:TEL:TRAC:CONF:RUN?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:SETUp:RUN</p> <p>SOURce[1..n]:DATA:TELEcom:PING:SETUp:RUN?</p>

---

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:TRACe:CONFIg:RUN?**

<b>Description</b>	<p>This query returns the status of the Trace command.</p> <p>This query is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Trace Route</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TRACe:CONFIg:RUN?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Trace command.</p> <p>0, returns the status of the Trace command as OFF.</p> <p>1, returns the status of the Trace command as ON.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAC:CONF:RUN ON</p> <p>SOUR:DATA:TEL:TRAC:CONF:RUN?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:SETUp:RUN</p> <p>SOURce[1..n]:DATA:TELEcom:PING:SETUp:RUN?</p>

---

**:SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:TOUT**

<b>Description</b>	<p>This command allows you to enter the maximum time allowed between an ICMP echo and response. Choices are 200 ms to 10000 ms.</p> <p>At *RST condition, this value is set to 4000 ms.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Trace Route</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:TOUT &lt;wsp&gt; &lt;TOUT&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>TOUT:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the time out value.</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:TRAC:CONF:TOUT 10000 SOUR:DATA:TEL:TRAC:CONF:TOUT? Returns: 10000</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOUT SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOUT?</pre>

---

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:TOUT?**

<b>Description</b>	<p>This query returns the time allowed between an ICMP echo and response. Choices are 200 ms to 10000 ms.</p> <p>At *RST condition, this value is set to 4000 ms.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Trace</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:TOUT?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current time allowed between an ICMP echo and response is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the time out value.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAC:CONF:TOUT 10000</p> <p>SOUR:DATA:TEL:TRAC:CONF:TOUT?</p> <p>Returns: 10000</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOUT</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TOUT?</p>

**:SOURce[1..n]:DATA:TELEcom:TRACe:CONFIg:HCOunt**

<b>Description</b>	<p>This command allows you to enter the number of network device packets.</p> <p>At *RST condition, this value is set to 128.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Trace Route</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TRACe:CONFIg:HCOunt &lt;wsp&gt; &lt;HOP&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>HOP:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Trace maximum hop count.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAC:CONF:HCO 88</p> <p>SOUR:DATA:TEL:TRAC:CONF:HCO?</p> <p>Returns: 88</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFIg:TTL</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFIg:TTL?</p>

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:HCOunt?**

<b>Description</b>	<p>This query returns the maximum network device packets. Choices are from 1 to 255.</p> <p>At *RST condition, this value is set to 128.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Trace Route</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:HCOunt?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current network device packets value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the trace maximum hop count.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:TRAC:CONF:HCO 88</p> <p>SOUR:DATA:TEL:TRAC:CONF:HCO?</p> <p>Returns: 88</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TTL</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:TTL?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:CONTInuous**

<b>Description</b>	<p>This command sets the Ping attempts to be made as continuous or Fixed (nAttempts). At *RST condition, this value is set to Continuous. Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping (Attempts dropdown)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFig:CONTInuous <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>SET:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element. The allowed elements for this parameter are: ON   OFF Sets the attempts.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:CONT ON SOUR:DATA:TEL:PING:CONF:CONT? Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:RUN SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:RUN?</p>

---

## SCPI Command Reference

### *Ping & Trace Route*

---

#### :SOURce[1..n]:DATA:TELEcom:PING:CONFig:CONTInuous?

<b>Description</b>	<p>This query returns status of Ping attempts.</p> <p>At *RST condition, this value is set to Continuous.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping (Attempts dropdown)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFig:CONTInuous?
<b>Response Syntax</b>	<Continuous>
<b>Response(s)</b>	<p><b>Continuous:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Ping attempts as continuous or Fixed (nAttempts).</p> <p>1, returns the attempts as continuous.</p> <p>0, returns the attempts as Fixed (nAttempts).</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:CONT ON</p> <p>SOUR:DATA:TEL:PING:CONF:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:RUN</p> <p>SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:RUN?</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ATTempts**

---

<b>Description</b>	<p>This command sets the number of attempts of ping requests. Choices are 1 to 100.</p> <p>At *RST condition, this value is set to 4.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ATTempts &lt;wsp&gt; &lt;Attempts&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Attempts:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value for the number of attempts.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SOUR:DATA:TEL:PING:CONF:ATT 100 SOUR:DATA:TEL:PING:CONF:ATT? Returns: 100</pre>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:PING:CONFig:CONTinuous</code>

---

## SCPI Command Reference

### *Ping & Trace Route*

---

---

#### :SOURce[1..n]:DATA:TELEcom:PING:CONFig:ATTempts?

---

<b>Description</b>	<p>This query returns the number of attempts of ping. Choices are 1 to 100.</p> <p>At *RST condition, this value is set to 4.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Ping</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ATTempts?[ <wsp>MAXimum   MINimum]
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current number of attempts of ping is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of attempts of ping requests.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:ATT 100</p> <p>SOUR:DATA:TEL:PING:CONF:ATT?</p> <p>Returns: 100</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PING:CONFig:CONTinuous?

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:SOURce:IP?**

<b>Description</b>	<p>This query returns the ping IP Source address.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path: Test &gt; IPV4&gt;Functions &gt; Ping &amp; Trace Route &gt; Source IP Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:SOURce:IP?
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the ping IPV4 source address in the form of a string.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:ADDR:SOUR:IP?</p> <p>Returns: "230.170.18.19"</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TRACe:CONFig:RUN?

---

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP**

<b>Description</b>	<p>This command sets the ping IP destination address.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Destination IP Address</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP &lt;wsp&gt; &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Set the ping IP address of the destination.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:ADDR:DEST:IP "230.170.18.19"</p> <p>SOUR:DATA:TEL:PING:CONF:ADDR:DEST:IP?</p> <p>Returns: "230.170.18.19"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP?</p>

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFIg:ADDRess:DESTination:IP?**

<b>Description</b>	<p>This query returns the ping IP destination address.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Destination IP Address</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFIg:ADDRess:DESTination:IP?
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the ping IP destination address in the form of a string.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:ADDR:DEST:IP "230.170.18.19"</p> <p>SOUR:DATA:TEL:PING:CONF:ADDR:DEST:IP?</p> <p>Returns: "230.170.18.19"</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:ADDRess:DESTination:IP?</p>

---

## SCPI Command Reference

### *Ping & Trace Route*

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP:USTReam**

<b>Description</b>	<p>This command sets the stream for ping IP destination address.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Destination IP Address &gt; Use Stream</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP:USTReam &lt;wsp&gt;ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the stream status for ping IP destination address.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:ADDR:DEST:IP:USTR ON</p> <p>SOUR:DATA:TEL:PING:CONF:ADDR:DEST:IP:USTR?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP?</p>

---

---

**:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP:USTReam?**

---

<b>Description</b>	<p>This returns the status of the stream for ping IP destination address.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Ping &amp; Trace Route &gt; Destination IP Address &gt; Use Stream</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP:USTReam?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the stream for ping IP destination address.</p> <p>1, returns the stream as ON.</p> <p>0, returns the stream as OFF.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:PING:CONF:ADDR:DEST:IP:USTR ON</p> <p>SOUR:DATA:TEL:PING:CONF:ADDR:DEST:IP:USTR?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP</p> <p>SOURce[1..n]:DATA:TELEcom:PING:CONFig:ADDRess:DESTination:IP?</p>

---

## Filters

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer**

<b>Description</b>	<p>This command enables or disables the corresponding filter for a specific Filter Number.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Enable</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer <wsp> <Filterno>, ON   OFF
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>SET:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the corresponding filter for a specific filter number.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT 1, ON</p> <p>SENS:DATA:TEL:ETH:STR:FILT? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:DESTinaton:PORT?



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer?**

<b>Description</b>	<p>The query returns the status of corresponding filter for a specific Filter Number.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Enable</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer? <wsp> <Filterno>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of corresponding filter for a specific filter number.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT 1, ON</p> <p>SENS:DATA:TEL:ETH:STR:FILT? 1</p> <p>Returns: 1</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:FNUMber?

---

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE**

**Description**

This command selects the type of filter.

At \*RST condition, this value is set to None.

Navigation Path: Functions > Filters & Packet Capture > Filter CONFig > Filter

**Syntax**

:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE <wsp> <Filterno>, <Criterion>, NONE | DMAC | SMAC | VLANid | VPriority | VLAN2id | V2Priority | VLAN3id | V3Priority | IPDestination | IPSource | TOS | PRECedence | DSErviceS | UDESTination | USOurce | ETHertype | IPPRotocol | IPVDESTINATION | IPVSOURCE | IPVFLABEL | IPVNHEADER | IPVDIFFSERV | IPVPRECEDENCE | IPVTON | MLABEL1 | MLABEL2 | MCOS1 | MCOS2

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE**

**Parameter(s)****Filterno:**

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the filter number from 1 to 10.

**Criterion:**

The program data syntax for the second parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the filter criterion number from 1 to 4.

**Type:**

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed elements for this parameter are: NONE | DMAC | SMAC | VLANid | VPRiority | VLAN2id | V2PRiority | VLAN3id | V3PRiority | IPDestination | IPSource | TOS | PRECedence | DSERvices | UDEStination | USOurce | ETHertype | IPPRotocol | IPVDESTINATION | IPVSOURCE | IPVFLABEL | IPVNHEADER | IPVDIFFSERV | IPVPRECEDENCE | IPV TOS | MLABEL1 | MLABEL2 | MCOS1 | MCOS2

Selects the type of filter.

NONE: No filter; DMAC: Destination MAC address; SMAC: the Source MAC address; VLANid: C-VLAN ID; VPRiority: C-VLAN Priority; VLAN2id: S-VLAN ID; V2PRiority: S-VLAN Priority; VLAN3id: E-VLAN ID; V3PRiority: E-VLAN Priority; IPDestination: the Destination IP address; IPSource: the IP Source address; TOS: Type of Service (TOS); PRECedence: Precedence; DSERvices: Differentiated Services; UDEStination: UDP Destination port; USOurce: UDP Source port; ETHertype: Ether Type; IPPRotocol: IP Protocol; IPVDESTINATION: IPv6 Destination IP address; IPVSOURCE: IPv6 Source address; IPVFLABEL: IPv6 Flow Label; IPVNHEADER: IPv6 Next Header; IPVDIFFSERV: IPv6 Differentiated Services; IPVPRECEDENCE: IPv6 Precedence; IPV TOS: IPv6 Type of Service (TOS); MLABEL1: MPLS Label 1; MLABEL2: MPLS Label 2; MCOS1: MPLS COS 1; MCOS2: MPLS COS 2

**Example(s)**

SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, DMAC

SENS:DATA:TEL:ETH:STR:FILT:TYPE? 1, 1

Returns: DMAC

**See Also**

SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE

---

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE?

---

<b>Description</b>	<p>This query returns the type of filter.</p> <p>At *RST condition, this value is set to None.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filter CONFig &gt; Filter</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<p>&lt;Type&gt;</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE?**
**Response(s)****Type:**

The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element.

Returns the type of filter.

NONE: No filter; DMAC: Destination MAC address; SMAC: the Source MAC address; VLANid: C-VLAN ID; VPriority: C-VLAN Priority; VLAN2id: S-VLAN ID; V2Priority: S-VLAN Priority; VLAN3id: E-VLAN ID; V3Priority: E-VLAN Priority; IPDestination: the Destination IP address; IPSource: the IP Source address; TOS: Type of Service (TOS); PRECedence: Precedence; DSERvices: Differentiated Services; UDESTination: UDP Destination port; USOurce: UDP Source port; ETHertype: Ether Type; IPPRotocol: IP Protocol; IPVDESTINATION: IPv6 Destination IP address; IPVSOURCE: IPv6 Source address; IPVFLABEL: IPv6 Flow Label; IPVNHEADER: IPv6 Next Header; IPVDIFFSERV: IPv6 Differentiated Services; IPVPRECEDENCE: IPv6 Precedence; IPVTOS, indicates IPv6 Type of Service (TOS);MLABEL1: MPLS Label 1; MLABEL2: MPLS Label 2; MCOS1: MPLS COS 1; MCOS2: MPLS COS 2

**Example(s)**

SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, DMAC

SENS:DATA:TEL:ETH:STR:FILT:TYPE? 1, 1

Returns: DMAC

**See Also**

SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE

---

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP

<b>Description</b>	<p>This command sets the Destination Internet Protocol (IP) address, if the filter type is IP Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Destination IP &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP <wsp> <Filterno>, <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Destination IP address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPD</p> <p>SENS:DATA:TEL:ETH:STR:FILT:DEST:IP 2,1, "0.1.1.1"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:DEST:IP? 2,1</p> <p>Returns: "0.1.1.1"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC?</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP?**

<b>Description</b>	<p>This query returns the Destination Internet Protocol (IP) address, if filter the type is IP Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Destination IP &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP? <wsp><Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns destination IP address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPD</p> <p>SENS:DATA:TEL:ETH:STR:FILT:DEST:IP 2,1, "0.1.1.1"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:DEST:IP? 2,1</p> <p>Returns: "0.1.1.1"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC?</p>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FRAMe:FORMat

---

<b>Description</b>	<p>Navigation Path: Functions-&gt;Filters &amp; Capture -&gt;Filters Configuration -&gt; Value</p> <p>This command sets the Destination Internet Protocol (IP) address, if filter type is IP address for a specific Filter No.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FRAMe:FORMat &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, ETHERNETII   8023SNAP   8023LLC</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b> The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element. Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b> The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element. Selects the filter criterion number from 1 to 4.</p> <p><b>Format:</b> The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element. The allowed elements for this parameter are: ETHERNETII   8023SNAP   8023LLC Selects the frame format.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, FRAMeformat SENS:DATA:TEL:ETH:STR:FILT:FRAM:FORM 2,1, 8023LLC SENS:DATA:TEL:ETH:STR:FILT:FRAM:FORM? 2,1 Returns: 8023LLC</pre>
<b>See Also</b>	<pre>LINS8599:SENSe:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE 1, 1, DMAC LINS8599:SENSe:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE? 1, 1</pre>

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FRAMe:FORMat?**

<b>Description</b>	<p>Navigation Path:</p> <p>Functions-&gt;Filters &amp; Capture -&gt;Filters Configuration -&gt; Value</p> <p>This query returns the Destination Internet Protocol (IP) address, if filter type is IP address for a specific Filter No.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FRAMe:FORMat? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns destination frame format.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, FRAMEformat</p> <p>SENS:DATA:TEL:ETH:STR:FILT:FRAM:FORM 2,1, 8023LLC</p> <p>SENS:DATA:TEL:ETH:STR:FILT:FRAM:FORM? 2,1</p> <p>Returns: 8023LLC</p>
<b>See Also</b>	<p>LINS8599:SENSe:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE 1, 1, DMAC</p> <p>LINS8599:SENSe:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE? 1, 1</p>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC

<b>Description</b>	<p>This command sets the Destination Media Access Control (MAC) address, if the the filter type is MAC Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""00:00:00:00:00:00"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type MAC Destination Address &gt; Value</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Address&gt;</code>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Destination MAC address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, DMAC SENS:DATA:TEL:ETH:STR:FILT:DEST:MAC 2,1, "00:00:00:FF:FF:FF" SENS:DATA:TEL:ETH:STR:FILT:DEST:MAC? 2,1 Returns: "00:00:00:FF:FF:FF"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP?</pre>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC?**

<b>Description</b>	<p>This query returns the Destination Media Access Control (MAC) address, if the filter type is MAC Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""00:00:00:00:00:00"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type MAC Destination Address &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC? <wsp><Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns destination MAC address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, DMAC</p> <p>SENS:DATA:TEL:ETH:STR:FILT:DEST:MAC 2,1, "00:00:00:FF:FF:FF"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:DEST:MAC? 2,1</p> <p>Returns: "00:00:00:FF:FF:FF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP?</p>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP

---

<b>Description</b>	<p>This command sets the User Datagram Protocol (UDP) Destination Port, if the filter type is UDP Destination Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type UDP Destination Port &gt; Value</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;port&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>port:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Destination UDP port address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, UDES SENS:DATA:TEL:ETH:STR:FILT:DEST:UDP 2,1, 50 SENS:DATA:TEL:ETH:STR:FILT:DEST:UDP? 2,1 Returns: 50</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC</pre>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?**

<b>Description</b>	<p>This query returns the User Datagram Protocol (UDP) Destination Port, if the filter type is UDP Destination Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type UDP Destination Port &gt; Value</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;[, MAXimum   MINimum]</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current UDP Destination Port for a specific Filter Number is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<pre>&lt;port&gt;</pre>

---

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?

<b>Response(s)</b>	<b>port:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns destination UDP port address.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, UDES SENS:DATA:TEL:ETH:STR:FILT:DEST:UDP 2,1, 50 SENS:DATA:TEL:ETH:STR:FILT:DEST:UDP? 2,1 Returns: 50
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERVICES**

<b>Description</b>	<p>This command sets the Differentiated Services, if the filter type is Diff Serv for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Diff Serv &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERVICES <wsp> <Filterno>, <Criterion>, <Dservices>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Differentiated Services.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, DSER SENS:DATA:TEL:ETH:STR:FILT:DSER 2,1, 55 SENS:DATA:TEL:ETH:STR:FILT:DSER? 2,1 Returns: 55</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECedence SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECedence?</pre>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERVICES?

---

<b>Description</b>	<p>This query returns the Differentiated Services, if the filter type is Diff Serv for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Diff Serv &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERVICES? <wsp> <Filterno>, <Criterion>[, <Dservices>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Differentiated Services.</p> <p>This parameter is optional. If no token is specified, the current Diff Serv for a specific Filter Number is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Dservices>

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices?**

<b>Response(s)</b>	<b>Dservices:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of differentiated services.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, DSER SENS:DATA:TEL:ETH:STR:FILT:DSER 2,1, 50 SENS:DATA:TEL:ETH:STR:FILT:DSER? 2,1 Returns: 50
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECedence SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECedence?

---

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECedence

<b>Description</b>	<p>This command sets the Precedence, if the filter type is Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Precedence &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECedence <wsp> <Filterno>, <Criterion>, <Precedence>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the Precedence.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, PREC SENS:DATA:TEL:ETH:STR:FILT:PREC 2,1, 4 SENS:DATA:TEL:ETH:STR:FILT:PREC? 2,1</pre> <p>Returns: 4</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices?</pre>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:PRECedence?**

<b>Description</b>	<p>This query returns the value of Precedence, if filter type is Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Precedence &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:PRECedence? <wsp> <Filterno>, <Criterion>[, <Precedence>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Precedence.</p> <p>This parameter is optional. If no token is specified, the current precedence value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Precedence>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PREcedence?

<b>Response(s)</b>	<b>Precedence:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of precedence.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, PREC SENS:DATA:TEL:ETH:STR:FILT:PREC 2,1, 2 SENS:DATA:TEL:ETH:STR:FILT:PREC? 2,1 Returns: 2
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP**

<b>Description</b>	<p>This command sets the source IP address, if the filter type is IP Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path: Functions &gt; Filters &gt; Filters &amp; Packet Capture &gt; Filter Configuration &gt; Filter type IP Source Address &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP <wsp> <Filterno>, <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Source IP address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPS</p> <p>SENS:DATA:TEL:ETH:STR:FILT:SOUR:IP 2,1, "0.1.1.1"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:SOUR:IP? 2,1</p> <p>Returns: "0.1.1.1"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC?</p>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP?

---

<b>Description</b>	<p>This query returns the Source IP address, if the filter type is IP Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0.0.0.0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filters &amp; Packet Capture &gt; Filter type IP Address &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns source IP address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPS</p> <p>SENS:DATA:TEL:ETH:STR:FILT:SOUR:IP 2,1, "0.1.1.1"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:SOUR:IP? 2,1</p> <p>Returns: "0.1.1.1"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC**

<b>Description</b>	<p>This command sets the Source MAC address, if the filter type is MAC Address Source for specific Filter Number.</p> <p>At *RST condition, this value is set to ""00:00:00:00:00:00"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type MAC Source Address &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC <wsp><Filterno>, <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Source MAC address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, SMAC</p> <p>SENS:DATA:TEL:ETH:STR:FILT:SOUR:MAC 2,1, "00:00:00:FF:FF:FF"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:SOUR:MAC? 2,1</p> <p>Returns: "00:00:00:FF:FF:FF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP?</p>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC?

<b>Description</b>	<p>This query returns the Source MAC address, if the filter type is MAC address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""00:00:00:00:00:00"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type MAC Source Address &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns source MAC address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 3,1, SMAC</p> <p>SENS:DATA:TEL:ETH:STR:FILT:SOUR:MAC 3,1, "00:00:00:FF:FF:FF"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:SOUR:MAC? 3,1</p> <p>Returns: "00:00:00:FF:FF:FF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP?</p>



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:UDP**

<b>Description</b>	<p>This command sets the User Data Protocol (UDP) Source Port, if the filter type is UDP Source Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters Configuration &gt; Filter type UDP Source Port &gt; Value</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:UDP &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;port&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>port:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Source UDP Port address.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, USO SENS:DATA:TEL:ETH:STR:FILT:SOUR:UDP 2,1, 65 SENS:DATA:TEL:ETH:STR:FILT:SOUR:UDP? 2,1 Returns: 65</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP</pre>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:UDP?

---

<b>Description</b>	<p>This query returns the User Data Protocol (UDP) source port, if the Filter type is UDP source port for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type UDP Source Port &gt; Value</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:UDP? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;[, MAXimum   MINimum]</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Source UDP Port address.</p> <p>This parameter is optional. If no token is specified, the current UDP source port for a specific Filter Number is returned.</p> <p>MAXimum: Biggest supported value MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<pre>&lt;port&gt;</pre>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:UDP?**

<b>Response(s)</b>	<b>port:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns source UDP port address.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, USO SENS:DATA:TEL:ETH:STR:FILT:SOUR:UDP 2,1, 65 SENS:DATA:TEL:ETH:STR:FILT:SOUR:UDP? 2,1 Returns: 65
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP?

---

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS

<b>Description</b>	<p>This command sets the Type of Service (TOS) value, if the filter type is TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0*00"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filter Configuration &gt; Filter type TOS &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS <wsp> <Filterno>, <Criterion>, <Tos>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the TOS value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, TOS</p> <p>SENS:DATA:TEL:ETH:STR:FILT:TOS 2,1, 30</p> <p>SENS:DATA:TEL:ETH:STR:FILT:TOS? 2,1</p> <p>Returns: 30</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS?**

<b>Description</b>	<p>This query returns the Type of Service (TOS) value, if the filter type is TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0*00"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filter type TOS &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS? <wsp> <Filterno>, <Criterion>[, <Tos>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Gets the TOS value.</p> <p>This parameter is optional. If no token is specified, the current TOS value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Tos>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS?

<b>Response(s)</b>	<b>Tos:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of the Type of Service (TOS) value.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, TOS SENS:DATA:TEL:ETH:STR:FILT:TOS 2,1, 30 SENS:DATA:TEL:ETH:STR:FILT:TOS? 2,1 Returns: 30
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID**

---

**Description**

This command sets the value of the VLAN identifier, if the filter type is VLAN ID for a specific Filter Number.

At \*RST condition, this value is set to ""0"".

Navigation Path: Functions > Filters & Packet Capture > Filters > Filter Configuration > Filter type Vlan ID > Value

**Syntax**

:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID <wsp> <Filterno>, <Criterion>, <Vlan>, <Id> | MAXimum | MINimum

---

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID**

<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Sets the VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p><b>Id:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value of VLAN Identifier.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1,VLANID SENS:DATA:TEL:ETH:STR:FILT:VLAN:ID 2,1,1,4095 SENS:DATA:TEL:ETH:STR:FILT:VLAN:ID? 2,1,1 Returns: 4095</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRIority SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRIority?</pre>

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?**

<b>Description</b>	<p>This query returns the value of VLAN ID, if the filter type is VLAN ID for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""0"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Vlan ID &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID? <wsp> <Filterno>, <Criterion>, <Vlan>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Sets the VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p>The program data syntax for the fourth parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current VLAN value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Id>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?**

<b>Response(s)</b>	<b>Id:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of the VLAN identifier.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, VLANID SENS:DATA:TEL:ETH:STR:FILT:VLAN:ID 2,1, 1,4095 SENS:DATA:TEL:ETH:STR:FILT:VLAN:ID? 2,1, 1 Returns: 4095
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRIOriety**

<b>Description</b>	<p>This command sets the value of the VLAN priority, if the filter type is VLAN priority for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Vlan Priority &gt; Value</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRIOriety &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Vlan&gt;, &lt;Priority&gt;   MAXimum   MINimum</p>

---

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority

<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Sets the VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p><b>Priority:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value of VLAN Priority.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1,VPriority SENS:DATA:TEL:ETH:STR:FILT:VLAN:PRI 2,1,1,7 SENS:DATA:TEL:ETH:STR:FILT:VLAN:PRI? 2,1,1 Returns: 7</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?</pre>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRIority?**

<b>Description</b>	<p>This query returns the value of Virtual Local Area Network (VLAN) priority, if the filter type is VLAN priority for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Vlan Priority &gt; Value</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRIority? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Vlan&gt;[, MAXimum   MINimum]</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Sets the VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p>The program data syntax for the fourth parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value of VLAN Priority.</p> <p>This parameter is optional. If no token is specified, the current VLAN Priority value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<pre>&lt;Priority&gt;</pre>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority?

<b>Response(s)</b>	<b>Priority:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of VLAN priority.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1,VPriority SENS:DATA:TEL:ETH:STR:FILT:VLAN:PRI 2,1,1,7 SENS:DATA:TEL:ETH:STR:FILT:VLAN:PRI? 2,1,1 Returns: 7
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol**

<b>Description</b>	<p>This command sets the IP Protocol value, if the filter type is IP Protocol for a specific Filter Number.</p> <p>At *RST condition, this value is set to 17.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type IP Protocol &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol <wsp> <Filterno>, <Criterion>, <Ipprotocol>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Ipprotocol:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the filter IP protocol value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1,1,IPPR SENS:DATA:TEL:ETH:STR:FILT:IPPR 1,1,17 SENS:DATA:TEL:ETH:STR:FILT:IPPR? 1, 1</pre> <p>Returns: 17</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype?</pre>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol?

<b>Description</b>	<p>This query returns the IP Protocol value, if the filter type is IP Protocol for a specific Filter Number.</p> <p>At *RST condition, this value is set to 17.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type IP Protocol &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol? <wsp> <Filterno>, <Criterion>[, <Ipprotocol>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Ipprotocol:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Gets the filter IP protocol value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Ipprotocol>

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol?**

<b>Response(s)</b>	<b>Ipprotocol:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the IP Protocol.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 1,1,IPPR SENS:DATA:TEL:ETH:STR:FILT:IPPR 1, 1, 17 SENS:DATA:TEL:ETH:STR:FILT:IPPR? 1, 1 Returns: 17
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype?

---

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype

---

<b>Description</b>	<p>This command sets the Ether Type value, if the filter type is Ether Type for a specific Filter Number.</p> <p>At *RST condition, this value is set to 2048.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Ether Type &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype <wsp> <Filterno>, <Criterion>, <Type>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Type:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the filter ether type value.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, ETH SENS:DATA:TEL:ETH:STR:FILT:ETH 2,1, #H007 SENS:DATA:TEL:ETH:STR:FILT:ETH? 2,1</pre> <p>Returns: 7</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol?</pre>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype?**

<b>Description</b>	<p>This query returns the Ether Type value, if the filter type is Ether Type for a specific Filter Number.</p> <p>At *RST condition, this value is set to 2048.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Ether Type &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Ether Type filter value.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, ETH</p> <p>SENS:DATA:TEL:ETH:STR:FILT:ETH 2,1, #H007</p> <p>SENS:DATA:TEL:ETH:STR:FILT:ETH? 2,1</p> <p>Returns: 7</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol?</p>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP**

<b>Description</b>	<p>This command sets the Mask Destination IP address, if the filter type is IP Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""255.255.255.255"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type IP Destination Address &gt; Mask</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Mask Destination IP address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:IP 2,1, "255.255.255.255"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:IP? 2,1</p> <p>Returns: "255.255.255.255"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP?**

<b>Description</b>	<p>This query returns the Mask Destination IP address, if the filter type is IP Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""255.255.255.255"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type IP Destination Address &gt; Mask</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP? &lt;wsp&gt;&lt;Filterno&gt;, &lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;Address&gt;</code>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the masked Destination IP address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPD SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:IP 2,1, "255.255.255.255" SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:IP? 2,1 Returns: "255.255.255.255"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC?</pre>

---

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC**

<b>Description</b>	<p>This command sets the Mask Destination MAC address, if the filter type is MAC Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""FF:FF:FF:FF:FF:FF"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Destination MAC Address &gt; Mask</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Mask Destination MAC address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1,DMAC</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:MAC 2,1,"FF:FF:FF:FF:FF:FF"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:MAC? 2,1</p> <p>Returns: "FF:FF:FF:FF:FF:FF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC?**

<b>Description</b>	<p>This query returns the Mask Destination MAC address, if the filter type is MAC Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""FF:FF:FF:FF:FF:FF"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Destination MAC Address &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the masked Destination MAC address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, DMAC</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:MAC 2,1, "FF:FF:FF:FF:FF:FF"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:MAC? 2,1</p> <p>Returns: "FF:FF:FF:FF:FF:FF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP?</p>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:UDP**

**Description** This command sets the Mask UDP Destination Port, if the filter type is UDP Destination Port for a specific Filter Number.

At \*RST condition, this value is set to 65535.

Navigation Path: Functions > Filters & Packet Capture > Filters > Filter Configuration > Filter type UDP Destination Port > Mask

**Syntax** :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:UDP  
<wsp> <Filterno>, <Criterion>, <Address>

**Parameter(s)** **Filterno:**

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the filter number from 1 to 10.

**Criterion:**

The program data syntax for the second parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the filter criterion number from 1 to 4.

**Address:**

The program data syntax for the third parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Selects the Mask Destination UDP port address.

**Example(s)** SENS:DATA:TEL:ETH:STR:FILT:TYPE 1,1,UDES

SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:UDP 1, 1, #HBE

SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:UDP? 1, 1

Returns: 190

**See Also** SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP

SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:UDP?**

<b>Description</b>	<p>This query returns the Mask UDP Destination Port, if the filter type is UDP Destination Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to 65535.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type UDP Destination Port &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:UDP? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<port>
<b>Response(s)</b>	<p><b>port:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the masked Destination UDP Port address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1,1,UDES</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:UDP 1, 1, #HBE</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:UDP? 1, 1</p> <p>Returns: 190</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?</p>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERVICES

<b>Description</b>	<p>This command sets the Mask Differentiated Services, if the filter type is Diff Serv for a specific Filter Number.</p> <p>At *RST condition, this value is set to 63.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter type Diff Serv &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERVICES <wsp> <Filterno>, <Criterion>, <Dservices>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask Differentiated Services.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, DSER SENS:DATA:TEL:ETH:STR:FILT:MASK:DSER 2,1, 21 SENS:DATA:TEL:ETH:STR:FILT:MASK:DSER? 2,1 Returns: 21</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECedence SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECedence?</pre>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices?**

<b>Description</b>	<p>This query returns the Mask Differentiated Services, if the filter type is Diff Serv for a specific Filter Number.</p> <p>At *RST condition, this value is set to 63.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Filter Type Diff Serv &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices? <wsp><Filterno>, <Criterion>[, <Dservices>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask Differentiated Services.</p> <p>This parameter is optional. If no token is specified, the current Mask Differentiated Services value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Dservices>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERVICES?

<b>Response(s)</b>	<b>Dservices:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the masked Differentiated Services.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, DSER SENS:DATA:TEL:ETH:STR:FILT:MASK:DSER 2,1, 20 SENS:DATA:TEL:ETH:STR:FILT:MASK:DSER? 2,1 Returns: 20
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:UDP?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PREcedence**

<b>Description</b>	<p>This command sets the Mask Precedence, if the filter type is Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PREcedence &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Precedence&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask Precedence.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, PREC SENS:DATA:TEL:ETH:STR:FILT:MASK:PREC 2,1, 5 SENS:DATA:TEL:ETH:STR:FILT:MASK:PREC? 2,1</pre> <p>Returns: 5</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERVICES SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERVICES?</pre>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECedence?

<b>Description</b>	<p>This query returns the Mask Precedence, if the filter type is Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filter Configuration &gt; Filter type Precedence &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECedence? <wsp><Filterno>, <Criterion>[, <Precedence>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask Precedence.</p> <p>This parameter is optional. If no token is specified, the current Mask Precedence value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Precedence>

---

---

**:SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:MASK:PRECedence?**

<b>Response(s)</b>	<b>Precedence:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the masked Precedence.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, PREC SENS:DATA:TEL:ETH:STR:FILT:MASK:PREC 2,1, 6 SENS:DATA:TEL:ETH:STR:FILT:MASK:PREC? 2,1 Returns: 6
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:MASK:DSERVICES SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:MASK:DSERVICES?

---

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP

<b>Description</b>	<p>This command sets the Mask Source IP Address, if the filter type is IP Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""255.255.255.255"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP <wsp> <Filterno>, <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Mask Source IP address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPS SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:IP 2,1, "255.255.255.255" SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:IP? 2,1 Returns: "255.255.255.255"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC?</pre>



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP?**

<b>Description</b>	<p>This query returns the Mask Source IP Address, if the filter type is IP Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""255.255.255.255"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the masked Source IP address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPS</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:IP 2,1, "255.255.255.255"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:IP? 2,1</p> <p>Returns: "255.255.255.255"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC?</p>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC

---

<b>Description</b>	<p>This command sets the Mask Source MAC Address, if the filter type is MAC Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""FF:FF:FF:FF:FF:FF"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC <wsp> <Filterno>, <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Mask Source MAC address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, SMAC</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:MAC 2,1, "FF:FF:FF:FF:FF:FF"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:MAC? 2,1</p> <p>Returns: "FF:FF:FF:FF:FF:FF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP?</p>

---

---

```
:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC?
```

<b>Description</b>	<p>This query returns the Mask Source MAC Address, if the filter type is MAC Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to ""FF:FF:FF:FF:FF:FF"".</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<pre>&lt;Address&gt;</pre>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the masked Source MAC address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, SMAC SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:MAC 2,1, "FF:FF:FF:FF:FF:FF" SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:MAC? 2,1 Returns: "FF:FF:FF:FF:FF:FF"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP?</pre>

---

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:UDP

---

<b>Description</b>	<p>This command sets the Mask UDP Source Port, if the filter type is UDP Source Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to 65535.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:UDP &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Address&gt;</code>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Mask Source UDP Port address.</p>
<b>Example(s)</b>	<code>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, USO</code> <code>SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:UDP 2,1, #HCF</code>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</code> <code>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:UDP</code>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:UDP?**

<b>Description</b>	<p>The query returns the Mask UDP Source Port, if the filter type is UDP Source Port for specific Filter Number.</p> <p>At *RST condition, this value is set to 65535.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:UDP? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<port>
<b>Response(s)</b>	<p><b>port:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the masked Source UDP Port address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, USO</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:UDP 2,1, #HCF</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:UDP? 2,1</p> <p>Returns: 207</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:UDP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:UDP?</p>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS

<b>Description</b>	<p>This command sets the Mask TOS, if the filter type is TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Tos&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask Type of Service (TOS).</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, TOS SENS:DATA:TEL:ETH:STR:FILT:MASK:TOS 2,1, 60 SENS:DATA:TEL:ETH:STR:FILT:MASK:TOS? 2,1 Returns: 60</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS?</pre>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS?**

<b>Description</b>	<p>This query returns the Mask TOS, if the filter type is TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS? <wsp> <Filterno>, <Criterion>[, <Tos>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask Type of Service (TOS).</p> <p>This parameter is optional. If no token is specified, the current Mask Type of Service (TOS) value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Tos>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS?

---

<b>Response(s)</b>	<b>Tos:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the masked Type of Service (TOS).
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, TOS SENS:DATA:TEL:ETH:STR:FILT:MASK:TOS 2,1, 90 SENS:DATA:TEL:ETH:STR:FILT:MASK:TOS? 2,1 Returns: 90
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS?

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:ID**

<b>Description</b>	<p>This command sets the Mask VLAN ID, if the filter type is VLAN ID for a specific Filter Number.</p> <p>At *RST condition, this value is set to 4095.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:ID <wsp> <Filterno>, <Criterion>, <Vlan>, <Id>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the masked VLAN.</p> <p><b>Id:</b></p> <p>The program data syntax for the fourth parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the value of VLAN Identifier.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, VLAN SENS:DATA:TEL:ETH:STR:FILT:MASK:VLAN:ID 2,1, 1, #HAB SENS:DATA:TEL:ETH:STR:FILT:MASK:VLAN:ID? 2,1, 1 Returns: 171</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?</pre>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:ID?

---

<b>Description</b>	<p>This query returns the Mask VLAN ID, if the filter type is VLAN ID for a specific Filter Number.</p> <p>At *RST condition, this value is set to 4095.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:ID? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Vlan&gt;</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the masked VLAN.</p>
<b>Response Syntax</b>	<pre>&lt;Id&gt;</pre>

---

**:SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:MASK:VLAN:ID?**

<b>Response(s)</b>	<b>Id:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of masked VLAN identifier.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, VLAN SENS:DATA:TEL:ETH:STR:FILT:MASK:VLAN:ID 2,1, 1, #HAB SENS:DATA:TEL:ETH:STR:FILT:MASK:VLAN:ID? 2,1, 1 Returns: 171
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:MASK:VLAN:ID SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:MASK:VLAN:ID?

---

## SCPI Command Reference

### *Filters*

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:PRIOriTy**

---

<b>Description</b>	<p>This command sets the Mask VLAN Priority, if the filter type is VLAN Priority for a specific Filter Number.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:PRIOriTy &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Vlan&gt;, &lt;Priority&gt;   MAXimum   MINimum</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:PRiority****Parameter(s)****Filterno:**

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the filter number from 1 to 10.

**Criterion:**

The program data syntax for the second parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the filter criterion number from 1 to 4.

**Vlan:**

The program data syntax for the third parameter is defined as a <numeric\_value> element.

Selects the masked VLAN.

**Priority:**

The program data syntax for the fourth parameter is defined as a <numeric\_value> element.

The allowed elements for this parameter are: MAXimum | MINimum

Sets the value of VLAN Priority.

MAXimum: Biggest supported value

MINimum: Smallest supported value

**Example(s)**

SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, VPRiority

SENS:DATA:TEL:ETH:STR:FILT:MASK:VLAN:PRI 1, 1, 1, MAX

SENS:DATA:TEL:ETH:STR:FILT:MASK:VLAN:PRI? 1, 1, 1

Returns: 4095

**See Also**

SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE

SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority

SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority?

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:PRIOriTy?**

<b>Description</b>	<p>This query returns the Mask VLAN Priority, if the filter type is VLAN Priority for a specific Filter Number.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:PRIOriTy? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Vlan&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the masked VLAN.</p> <p>The program data syntax for the fourth parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current masked VLAN value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Priority&gt;</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:VLAN:PRiority?**

<b>Response(s)</b>	<b>Priority:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of masked VLAN Priority.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, VPRiority SENS:DATA:TEL:ETH:STR:FILT:MASK:VLAN:PRI 1, 1, 1, MAX SENS:DATA:TEL:ETH:STR:FILT:MASK:VLAN:PRI? 1, 1, 1 Returns: 4095
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority?

---

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:IPPRotocol

<b>Description</b>	<p>This command sets Mask IP Protocol, if the filter type is IP Protocol for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:IPPRotocol &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Ipprotocol&gt;</code>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Ipprotocol:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Mask IP protocol value.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1,IPPR SENS:DATA:TEL:ETH:STR:FILT:MASK:IPPR 2,1,#HFF SENS:DATA:TEL:ETH:STR:FILT:MASK:IPPR? 2,1 Returns: 255</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol?</pre>



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:IPPRotocol?**

<b>Description</b>	<p>This query returns the Mask IP Protocol, if the filter type is IP Protocol for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:IPPRotocol? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Ipprotocol>
<b>Response(s)</b>	<p><b>Ipprotocol:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Mask IP Protocol.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPPR</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:IPPR 2,1, #HFF</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:IPPR? 2,1</p> <p>Returns: 255</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol?</p>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:ETHertype

<b>Description</b>	<p>This command sets Mask Ether Type, if the filter type is Ether Type for a specific Filter Number.</p> <p>At *RST condition, this value is set to 65535.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:ETHertype <wsp> <Filterno>, <Criterion>, <Type>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Type:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Mask filter ether type value.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1,1,ETH SENS:DATA:TEL:ETH:STR:FILT:MASK:ETH 1,1,#HAA SENS:DATA:TEL:ETH:STR:FILT:MASK:ETH? 1,1</pre> <p>Returns: 170</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype?</pre>

**:SENSe[1..n]:DATA:TELeom:ETHernet:STReam:FILTer:MASK:ETHertype?**

<b>Description</b>	<p>This query returns the Mask Ether Type, if the filter type is Ether Type for a specific Filter Number.</p> <p>At *RST condition, this value is set to 65535.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:ETHernet:STReam:FILTer:MASK:ETHertype? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Mask EtherType.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1,1, ETH</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:ETH 1, 1, #HAA</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:ETH? 1, 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELeom:ETHernet:STReam:FILTer:ETHertype</p> <p>SENSe[1..n]:DATA:TELeom:ETHernet:STReam:FILTer:ETHertype?</p>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ENABled:TIME?

---

<b>Description</b>	<p>This query returns the time during which the filter is enabled.</p> <p>At *RST condition, this value is set to '--'.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Enabled Time</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ENABled:TIME? &lt;wsp&gt; &lt;Filterno&gt;</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p>
<b>Response Syntax</b>	<pre>&lt;Time&gt;</pre>
<b>Response(s)</b>	<p><b>Time:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the time during which the filter is enabled.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:ENAB:TIME? 1</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMing:RATE?</pre>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:BANDwidth?**

---

<b>Description</b>	<p>This command sets the frame bandwidth in megabit per second.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Statistics &gt; Ethernet BW (Mbit/s)</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:BANDwidth? &lt;wsp&gt; &lt;Filterno&gt;</code>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p>
<b>Response Syntax</b>	<code>&lt;Bandwidth&gt;</code>
<b>Response(s)</b>	<p><b>Bandwidth:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frame bandwidth.</p>
<b>Example(s)</b>	<code>SENS:DATA:TEL:ETH:FILT:FRAM:BAND? 1</code>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMing:RATE?</code>

---

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:RATE?

---

<b>Description</b>	<p>This query returns the frame rate in frames per second.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Statistics &gt; Frame Rate</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:RATE? <wsp> <Filterno>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p>
<b>Response Syntax</b>	<Rate>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frame rate.</p>
<b>Example(s)</b>	SENS:DATA:TEL:ETH:FILT:FRAM:RATE? 1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMing:UTILization?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:UTILization?**

<b>Description</b>	<p>This query returns the frame utilization in percentage.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Statistics &gt; Line Utilization</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:UTILization? <wsp> <Filterno>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p>
<b>Response Syntax</b>	<Utilization>
<b>Response(s)</b>	<p><b>Utilization:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frame utilization.</p>
<b>Example(s)</b>	SENS:DATA:TEL:ETH:FILT:FRAM:UTIL? 1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMing:RATE?

---

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:COUNT?

---

<b>Description</b>	<p>This query returns the number of frames matching the configured filter's criteria.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Statistics &gt; Frame Count</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMe:COUNT? <wsp> <Filterno>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p>
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<p><b>Count:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the number of valid and invalid frames received.</p>
<b>Example(s)</b>	SENS:DATA:TEL:ETH:FILT:FRAM:COUN? 1
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:FILTer:FRAMing:RATE?

---



**:SENSe[1..n]:DATA:TELeom:ETHernet:FILTer:STATistics?**

<b>Description</b>	<p>This query returns the Filter error statistics.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Statistics &gt; Error Count</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELeom:ETHernet:FILTer:STATistics? &lt;wsp&gt; &lt;Filterno&gt;, FCS   IPCHECKSUM   JABBER   RUNT   UDPCHECKSUM   TCPCHECKSUM   UNDERSIZED   OVERSIZED</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Error:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FCS   IPCHECKSUM   JABBER   RUNT   UDPCHECKSUM   TCPCHECKSUM   UNDERSIZED   OVERSIZED</p> <p>Selects the type of MAC (Media Access Control) error.</p> <p>FCS: FCS (Frame Check Sequence)</p> <p>IPCHECKSUM: IP checksum</p> <p>JABBER: Jabber/Giant</p> <p>RUNT: Runt</p> <p>TCPCHECKSUM: Tcpchecksum</p> <p>UDPCHECKSUM: Udpchecksum</p> <p>UNDERSIZED: Undersized</p> <p>OVERSIZED: Oversized</p>
<b>Response Syntax</b>	<p>&lt;Errors&gt;</p>

## SCPI Command Reference

### *Filters*

---

---

#### **:SENSe[1..n]:DATA:TELecom:ETHernet:FILTer:STATistics?**

<b>Response(s)</b>	<b>Errors:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of Filter errors.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:FILT:STAT? 1, FCS
<b>See Also</b>	SENSe[1..n]:DATA:TELecom:ETHernet:FILTer:FRAMing:UTILization?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator**

<b>Description</b>	<p>This command selects the logical operator (AND or OR) between two operands when more than two operands are used.</p> <p>At *RST condition, this value is set to AND.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Oper.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator <wsp> <Filterno>, <Criterion>, AND   OR
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Operator:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AND   OR</p> <p>Selects the logical operators.</p> <p>AND</p> <p>OR</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:OPER 1, 1, AND</p> <p>SENS:DATA:TEL:ETH:STR:FILT:OPER? 1, 1</p> <p>Returns: AND</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator?</p>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator?

---

<b>Description</b>	<p>This query returns the logical operator (AND or OR) between two operands when more than two operands are used.</p> <p>At *RST condition, this value is set to AND.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Oper.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Operator>
<b>Response(s)</b>	<p><b>Operator:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the logical operators.</p> <p>AND, AND is selected as the logical operator.</p> <p>OR, OR is selected as the logical operator.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:OPER 2,1, AND</p> <p>SENS:DATA:TEL:ETH:STR:FILT:OPER? 2,1</p> <p>Returns: AND</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT**

<b>Description</b>	<p>This command selects the Operator Not. When it is selected, add the logical negation (not equal) operator for the operand filter defined at its right.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Not</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Set:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables the NOT operator.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:OPER:NOT 2,1, ON</p> <p>SENS:DATA:TEL:ETH:STR:FILT:OPER:NOT? 2,1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe?</p>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT?

<b>Description</b>	<p>This query returns the Operator Not. When selected, add the logical negation (not equal) operator for the operand filter defined at its right.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; Not</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;Set&gt;</code>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>This query returns the Operator Not.</p> <p>0, returns the status of the logical operator as OFF.</p> <p>1, returns the status of the logical operator as ON.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:OPER:NOT 2,1, ON SENS:DATA:TEL:ETH:STR:FILT:OPER:NOT? 2,1 Returns: 1</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe?</pre>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:OPEN**

<b>Description</b>	<p>This command selects the open parenthesis to control the precedence of operands when two operands are used.</p> <p>At *RST condition, this value is set to blank.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; (</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:OPEN <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Set:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables the open parenthesis.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:BRAC:OPEN 2,1, ON</p> <p>SENS:DATA:TEL:ETH:STR:FILT:BRAC:OPEN? 2,1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe?</p>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:OPEN?

<b>Description</b>	<p>This query returns the selected open parenthesis to control the precedence of operands when two operands are used.</p> <p>At *RST condition, this value is set to blank.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; (</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:OPEN? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<pre>&lt;Set&gt;</pre>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of open parenthesis.</p> <p>0, returns the status of open parenthesis as ON.</p> <p>1, returns the status of open parenthesis as OFF.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:BRAC:OPEN 2,1, ON SENS:DATA:TEL:ETH:STR:FILT:BRAC:OPEN? 2,1 Returns: 1</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe?</pre>



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe**

<b>Description</b>	<p>This command selects the close parenthesis to control the precedence of operands when two operands are used.</p> <p>At *RST condition, this value is set to blank.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; )</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Set:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables the close parenthesis.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:BRAC:CLOS 2,1, ON</p> <p>SENS:DATA:TEL:ETH:STR:FILT:BRAC:CLOS? 2,1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT?</p>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe?

---

<b>Description</b>	<p>This query returns the selected close parenthesis to control the precedence of operands when two operands are used.</p> <p>At *RST condition, this value is set to blank.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Filters &gt; Filter Configuration &gt; )</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of close parenthesis.</p> <p>0, returns the status of close parenthesis as OFF.</p> <p>1, returns the status of close parenthesis as ON.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:BRAC:CLOS 2,1, ON</p> <p>SENS:DATA:TEL:ETH:STR:FILT:BRAC:CLOS? 2,1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IPVersion**

<b>Description</b>	<p>This command sets the Destination IPv6 address, if Filter Type is IPv6 Destination address for a specific Filter No.</p> <p>At *RST condition, this value is set to ""0000:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IPVersion <wsp> <Filterno>, <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Destination IPv6 address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVD SENS:DATA:TEL:ETH:STR:FILT:DEST:IPV 1, 1, "0000:0000:0000:0000:0000:0000:0000:0000". SENS:DATA:TEL:ETH:STR:FILT:DEST:IPV? 1, 1 Returns: "0000:0000:0000:0000:0000:0000:0000:0000".</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERVICES:IPV?</pre>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IPVersion?

<b>Description</b>	<p>This query returns the Destination IPv6 address, if Filter Type is IPv6 Destination address for a specific Filter No.</p> <p>At *RST condition, this value is set to ""0000:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IPVersion? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Destination IPv6 address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVD</p> <p>SENS:DATA:TEL:ETH:STR:FILT:DEST:IPV 1, 1, "0000:0000:0000:0000:0000:0000:0000:0000".</p> <p>SENS:DATA:TEL:ETH:STR:FILT:DEST:IPV? 1, 1</p> <p>Returns: "0000:0000:0000:0000:0000:0000:0000:0000".</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPV</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPVersion**

<b>Description</b>	<p>This command sets the Source IPv6 address, if Filter Type is IPv6 Source address for a specific Filter No.</p> <p>At *RST condition, this value is set to ""0000:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt;&gt;Functions &gt;Filters &gt; Filter CONFIg &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPVersion <wsp><Filterno>, <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Source IPv6 address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVS SENS:DATA:TEL:ETH:STR:FILT:SOUR:IPV 1, 1, "0000:0000:0000:0000:0000:0000:0000:0000". SENS:DATA:TEL:ETH:STR:FILT:SOUR:IPV? 1, 1 Returns: "0000:0000:0000:0000:0000:0000:0000:0000".</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence:IPV?</pre>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPVersion?

---

<b>Description</b>	<p>This query returns the Source IPv6 address, if Filter Type is IPv6 Source address for a specific Filter No.</p> <p>At *RST condition, this value is set to ""0000:0000:0000:0000:0000:0000:0000:0000"".</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPVersion? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Source IPv6 address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVS</p> <p>SENS:DATA:TEL:ETH:STR:FILT:SOUR:IPV 1, 1, "0000:0000:0000:0000:0000:0000:0000:0000".</p> <p>SENS:DATA:TEL:ETH:STR:FILT:SOUR:IPV? 1, 1</p> <p>Returns: "0000:0000:0000:0000:0000:0000:0000:0000".</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECedence:IPV</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLABel:IPVersion**

<b>Description</b>	<p>This command sets the IPv6 Flow Label, if Filter Type is IPv6 Flow Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLABel:IPVersion <wsp> <Filterno>, <Criterion>, <Flabel>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Flabel:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the IPv6 Flow Label.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVF SENS:DATA:TEL:ETH:STR:FILT:FLAB:IPV 1, 1, 20 SENS:DATA:TEL:ETH:STR:FILT:FLAB:IPV? 1, 1 Returns: 20</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TCLass:IPV?</pre>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLABel:IPVersion?

<b>Description</b>	<p>This query returns the IPv6 Flow Label, if Filter Type is IPv6 Flow Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLABel:IPVersion? <wsp> <Filterno>, <Criterion>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current flow label is returned.</p>
<b>Response Syntax</b>	<Flabel>

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLABel:IPVersion?**

<b>Response(s)</b>	<b>Flabel:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the IPv6 Flow Label.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVF SENS:DATA:TEL:ETH:STR:FILT:FLAB:IPV 1, 1, 20 SENS:DATA:TEL:ETH:STR:FILT:FLAB:IPV? 1, 1 Returns: 20
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TCLass:IPV

---

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHEader:IPVersion

---

**Description** This command sets the IPv6 Next Header, if Filter Type is IPv6 Next Header for a specific Filter No.

At \*RST condition, this value is set to 0.

Navigation Path: Test > Functions > Filters > Filter CONFIG > Value

**Syntax** :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHEader:IPVersion <wsp>  
<Filterno>, <Criterion>, <Nheader> | MAXimum | MINimum

**Parameter(s)** **Filterno:**

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the filter number from 1 to 10.

**Criterion:**

The program data syntax for the second parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the filter criterion number from 1 to 4.

**Nheader:**

The program data syntax for the third parameter is defined as a <numeric\_value> element.

The allowed elements for this parameter are: MAXimum | MINimum

Sets the IPv6 Next Header.

**Example(s)** SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVN

SENS:DATA:TEL:ETH:STR:FILT:NHE:IPV 1, 1, 20

SENS:DATA:TEL:ETH:STR:FILT:NHE:IPV? 1, 1

Returns: 20

**See Also** SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IPV?

SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHEader:IPVersion?**

<b>Description</b>	<p>This query returns the IPv6 Next Header, if Filter Type is IPv6 Next Header for a specific Filter No.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHEader:IPVersion? <wsp><Filterno>, <Criterion>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current next header is returned.</p>
<b>Response Syntax</b>	<Nheader>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHEader:IPVersion?

<b>Response(s)</b>	<b>Nheader:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the IPv6 Next Header.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVN SENS:DATA:TEL:ETH:STR:FILT:NHE:IPV 1, 1, 20 SENS:DATA:TEL:ETH:STR:FILT:NHE:IPV? 1, 1 Returns: 20
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPVersion**

<b>Description</b>	<p>This command sets the value of Differentiated Services, if filter type is Differentiated Services for a specific Filter No.</p> <p>At *RST condition, this value is set to #B000000.</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPVersion <wsp> <Filterno>, <Criterion>, <Dservices>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the value of differentiated services.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1,IPVD SENS:DATA:TEL:ETH:STR:FILT:DSER:IPV 2,1,#B000001 SENS:DATA:TEL:ETH:STR:FILT:DSER:IPV? 2,1 Returns: 1</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPV? SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</pre>

## SCPI Command Reference

### Filters

---

---

#### **:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPVersion?**

---

<b>Description</b>	<p>This query returns the value of Differentiated Services, if filter type is Differentiated Services for a specific Filter No.</p> <p>At *RST condition, this value is set to #B000000.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPVersion? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;Dservices&gt;</code>
<b>Response(s)</b>	<p><b>Dservices:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of differentiated services.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1,IPVD SENS:DATA:TEL:ETH:STR:FILT:DSER:IPV 2,1,#B000001 SENS:DATA:TEL:ETH:STR:FILT:DSER:IPV? 2,1 Returns: 1</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</pre>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECedence:IPVersion**

<b>Description</b>	<p>This command sets the value of Precedence, if filter type is Precedence for a specific Filter No.</p> <p>At *RST condition, this value is set to #B000.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECedence:IPVersion <wsp> <Filterno>, <Criterion>, <Precedence>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Precedence.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPVP SENS:DATA:TEL:ETH:STR:FILT:PREC:IPV 2,1, #B001 SENS:DATA:TEL:ETH:STR:FILT:PREC:IPV? 2,1</pre> <p>Returns: 1</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLABel:IPV? SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</pre>

## SCPI Command Reference

### Filters

---

#### **:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence:IPVersion?**

<b>Description</b>	<p>This query returns the value of Precedence, if filter type is Precedence for a specific Filter No.</p> <p>At *RST condition, this value is set to #B000.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence:IPVersion? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;Precedence&gt;</code>
<b>Response(s)</b>	<p><b>Precedence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of precedence.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPVP SENS:DATA:TEL:ETH:STR:FILT:PREC:IPV 2,1, #B001 SENS:DATA:TEL:ETH:STR:FILT:PREC:IPV? 2,1 Returns: 1</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLABel:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</pre>



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS:IPVersion**

<b>Description</b>	<p>This command sets the Traffic Class (TOS/DS) value of IPv6.</p> <p>At *RST condition, this value is set to #B00.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS:IPVersion <wsp> <Filterno>, <Criterion>, <Tclass>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tclass:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Traffic Class (TOS/DS) value.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPV1 SENS:DATA:TEL:ETH:STR:FILT:TOS:IPV 2,1, #B001 SENS:DATA:TEL:ETH:STR:FILT:TOS:IPV? 2,1</pre> <p>Returns: 1</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TCLass:IPV?</pre>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS:IPVersion?

<b>Description</b>	<p>This query returns the Traffic Class (TOS/DS) value of IPv6.</p> <p>At *RST condition, this value is set to #B00.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS:IPVersion? <wsp> <Filterno>, <Criterion>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Precedence>
<b>Response(s)</b>	<p><b>Precedence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of precedence.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPVT</p> <p>SENS:DATA:TEL:ETH:STR:FILT:TOS:IPV 2,1, #B001</p> <p>SENS:DATA:TEL:ETH:STR:FILT:TOS:IPV? 2,1</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TCLass:IPV</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IPVersion**

<b>Description</b>	<p>This command sets the Mask Destination IPv6 address, if Filter Type is IPv6 Destination address for a specific Filter No.</p> <p>At *RST condition, this value is set to ""FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF"".</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IPVersion <wsp> <Filterno>, <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Mask Destination IPv6 address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVDESTINATION SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:IPV 1, 1, "FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF" SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:IPV? 1, 1 Returns: "FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IPV? SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</pre>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IPVersion?**

<b>Description</b>	<p>This query returns the Mask Destination IPv6 address, if Filter Type is IPv6 Destination address for a specific Filter No.</p> <p>At *RST condition, this value is set to ""FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF"".</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IPVersion? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<p>&lt;Address&gt;</p>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Mask Destination IPv6 address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVDESTINATION</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:IPV 1, 1, "FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF"</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:DEST:IPV? 1, 1</p> <p>Returns: "FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IPV</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IPVersion**

<b>Description</b>	<p>This command sets the Mask Source IPv6 address, if Filter Type is IPv6 Source address for a specific Filter No.</p> <p>At *RST condition, this value is set to ""FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF"".</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IPVersion <wsp> <Filterno>, <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Sets the Mask Source IPv6 address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVSource SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:IPV 1, 1, "FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF" SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:IPV? 1, 1 Returns: "FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IPV? SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</pre>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IPVersion?**

<b>Description</b>	<p>This query returns the Mask Source IPv6 address, if Filter Type is IPv6 Source address for a specific Filter No.</p> <p>At *RST condition, this value is set to ""FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF"".</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IPVersion? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<p>&lt;Address&gt;</p>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Mask Source IPv6 address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVSource SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:IPV 1, 1, "FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF" SENS:DATA:TEL:ETH:STR:FILT:MASK:SOUR:IPV? 1, 1 Returns: "FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:FLABel:IPVersion**

<b>Description</b>	<p>This command sets the IPv6 Mask Flow Label, if Filter Type is IPv6 Flow Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 1048575.</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:FLABel:IPVersion <wsp> <Filterno>, <Criterion>, <Flabel>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Flabel:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the IPv6 Mask Flow Label.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVFLABEL SENS:DATA:TEL:ETH:STR:FILT:MASK:FLAB:IPV 1, 1, 60 SENS:DATA:TEL:ETH:STR:FILT:MASK:FLAB:IPV? 1, 1 Returns: 60</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:FLABel:IPV? SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</pre>

## SCPI Command Reference

### Filters

---

---

#### **:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:FLABel:IPVersion?**

---

<b>Description</b>	<p>This query returns the IPv6 Mask Flow Label, if Filter Type is IPv6 Flow Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 1048575.</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:FLABel:IPVersion? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current label number is returned.</p>
<b>Response Syntax</b>	<p>&lt;Flabel&gt;</p>

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:FLABel:IPVersion?**

<b>Response(s)</b>	<b>Flabel:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the IPv6 Mask Flow Label.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVFLABEL SENS:DATA:TEL:ETH:STR:FILT:MASK:FLAB:IPV 1, 1, 30 SENS:DATA:TEL:ETH:STR:FILT:MASK:FLAB:IPV? 1, 1 Returns: 30
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:FLABel:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE

---

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHEader:IPVersion**

<b>Description</b>	<p>This command sets the IPv6 Mask Next Header, if Filter Type is IPv6 Next Header for a specific Filter No.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFIG &gt; MASK</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHEader:IPVersion &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Nheader&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Nheader:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the IPv6 Mask Next Header.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVNheader SENS:DATA:TEL:ETH:STR:FILT:MASK:NHE:IPV 1, 1, 60 SENS:DATA:TEL:ETH:STR:FILT:MASK:NHE:IPV? 1, 1 Returns: 60</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHEader:IPV? SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:NHEader:IPVersion?**

<b>Description</b>	<p>This query returns the IPv6 Mask Next Header, if Filter Type is IPv6 Next Header for a specific Filter No.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTER:MASK:NHEader:IPVersion? <wsp> <Filterno>, <Criterion>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current label number is returned.</p>
<b>Response Syntax</b>	<Nheader>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHEader:IPVersion?

<b>Response(s)</b>	<b>Nheader:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the IPv6 Mask Next Header.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, IPVNheader SENS:DATA:TEL:ETH:STR:FILT:MASK:NHE:IPV 1, 1, 60 SENS:DATA:TEL:ETH:STR:FILT:MASK:NHE:IPV? 1, 1 Returns: 60
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHEader:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices:IPVersion**

<b>Description</b>	<p>This command sets the Mask Differentiated Services, if filter type is Differentiated Services for a specific Filter No.</p> <p>At *RST condition, this value is set to 63.</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFIG &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices:IPVersion <wsp> <Filterno>, <Criterion>, <Dservices>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the IPv6 Mask Differentiated Services.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPVDiffserv SENS:DATA:TEL:ETH:STR:FILT:MASK:DSER:IPV 2,1, 60 SENS:DATA:TEL:ETH:STR:FILT:MASK:DSER:IPV? 2,1 Returns: 60</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices:IPV?</pre>

## SCPI Command Reference

### Filters

---

---

#### **:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices:IPVersion?**

---

<b>Description</b>	<p>This query returns the Mask Differentiated Services, if filter type is Differentiated Services for a specific Filter No.</p> <p>At *RST condition, this value is set to 63.</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices:IPVersion? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;[, MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current label number is returned.</p>
<b>Response Syntax</b>	<code>&lt;Dservices&gt;</code>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSErviceS:IPVersion?**

<b>Response(s)</b>	<b>Dservices:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of differentiated services.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPVDiffserv SENS:DATA:TEL:ETH:STR:FILT:MASK:DSER:IPV 2,1, 60 SENS:DATA:TEL:ETH:STR:FILT:MASK:DSER:IPV? 2,1 Returns: 60
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSErviceS:IPV

---

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECedence:IPVersion

---

<b>Description</b>	<p>This command sets the value of Mask Precedence, if filter type is Precedence for a specific Filter No.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECedence:IPVersion <wsp> <Filterno>, <Criterion>, <Precedence>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the IPv6 Mask Precedence.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPVPRECEDENCE SENS:DATA:TEL:ETH:STR:FILT:MASK:PREC:IPV 2,1, 5 SENS:DATA:TEL:ETH:STR:FILT:MASK:PREC:IPV? 2,1 Returns: 5</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECedence:IPV?</pre>

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECEdence:IPVersion?**

<b>Description</b>	<p>This query returns the value of Mask Precedence, if filter type is Precedence for a specific Filter No.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECEdence:IPVersion? <wsp> <Filterno>, <Criterion>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current label number is returned.</p>
<b>Response Syntax</b>	<Precedence>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECedence:IPVersion?**

<b>Response(s)</b>	<b>Precedence:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of Mask Precedence.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPVPRECEDENCE SENS:DATA:TEL:ETH:STR:FILT:MASK:PREC:IPV 2,1, 5 SENS:DATA:TEL:ETH:STR:FILT:MASK:PREC:IPV? 2,1 Returns: 5
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:PRECedence:IPV

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS:IPVersion**

<b>Description</b>	<p>This command sets the Mask Traffic Class (TOS/DS) value of IPv6.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &gt; Filter CONFIg &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS:IPVersion <wsp> <Filterno>, <Criterion>, <Tclass>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tclass:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the IPv6 Mask Traffic Class (TOS/DS) value.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPVTOS SENS:DATA:TEL:ETH:STR:FILT:MASK:TOS:IPV 2,1, 69 SENS:DATA:TEL:ETH:STR:FILT:MASK:TOS:IPV? 2,1 Returns: 69</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TCLass:IPV?</pre>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS:IPVersion?**

<b>Description</b>	<p>This query returns the Mask Traffic Class (TOS/DS) value of IPv6.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Test &gt; Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TOS:IPVersion? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current label number is returned.</p>
<b>Response Syntax</b>	<p>&lt;Tclass&gt;</p>
<b>Response(s)</b>	<p><b>Tclass:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of TClass.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:STR:FILT:TYPE 2,1, IPV TOS</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:TOS:IPV 2,1, 69</p> <p>SENS:DATA:TEL:ETH:STR:FILT:MASK:TOS:IPV? 2,1</p> <p>Returns: 69</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:TCLass:IPV</p>

<b>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IPV</b>	
<b>Description</b>	<p>This command sets the Destination Internet Protocol (IPv6) address, if the filter type is IPv6 Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0000:0000:0000:0000:0000:0000:0000:0000.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IPV <wsp> <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Destination IPv6 address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:DEST:IPV 1, "0000:0000:0000:0000:0000:0000:0000:0000" SENS:DATA:TEL:ETH:FMAT:FILT:DEST:IPV? 1 Returns: "0000:0000:0000:0000:0000:0000:0000:0000"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOUR:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOUR:IPV?</pre>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IPV?

---

<b>Description</b>	<p>This query returns the Destination Internet Protocol (IPv6) address, if the filter type is IPv6 Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0000:0000:0000:0000:0000:0000:0000:0000.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFIg &gt; Filter CONFIg</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IPV? <wsp> <Criterion>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns destination IPv6 address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:DEST:IPV 1, "0000:0000:0000:0000:0000:0000:0000:0000"</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:DEST:IPV? 1</p> <p>Returns: "0000:0000:0000:0000:0000:0000:0000:0000"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:SOUR:IPV</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:SOUR:IPV?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERVICES:IPV**

<b>Description</b>	<p>This command sets the IPv6 Differentiated Services, if the filter type is IPv6 Diff Serv for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERVICES:IPV &lt;wsp&gt; &lt;Criterion&gt;, &lt;Dservices&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the IPv6 Differentiated Services.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:DSER:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:DSER:IPV? 1 Returns: 23</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence:IPV?</pre>

---

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERVICES:IPV?

<b>Description</b>	<p>This query returns the IPv6 Differentiated Services, if the filter type is IPv6 Diff Serv for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERVICES:IPV? <wsp> <Criterion>[, <Dservices>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the IPv6 Differentiated Services.</p> <p>This parameter is optional. If no token is specified, the current Dservices value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Dservices>
<b>Response(s)</b>	<p><b>Dservices:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of IPv6 differentiated services.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:DSER:IPV 1, 23</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:DSER:IPV? 1</p> <p>Returns: 23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:PRECEdence:IPV</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:PRECEdence:IPV?</p>



<b>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECedence:IPV</b>	
<b>Description</b>	<p>This command sets the IPv6 Precedence, if the filter type is IPv6Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECedence:IPV <wsp> <Criterion>, <Precedence>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the IPv6 Precedence.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:PREC:IPV 1, 23</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:PREC:IPV? 1</p> <p>Returns: 23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPV</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPV?</p>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECedence:IPV?**

<b>Description</b>	<p>This query returns the value of IPv6 Precedence, if the filter type is IPv6Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECedence:IPV? &lt;wsp&gt; &lt;Criterion&gt;[, &lt;Precedence&gt;   MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the IPv6 Precedence.</p> <p>This parameter is optional. If no token is specified, the current precedence value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Precedence&gt;</p>
<b>Response(s)</b>	<p><b>Precedence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of IPv6 precedence.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:PREC:IPV 1, 23</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:PREC:IPV? 1</p> <p>Returns: 23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPV</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices:IPV?</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:NHEader:IPV**

<b>Description</b>	<p>This command sets the IPv6 Next Header, if the filter type is IPv6 Next Header for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:NHEader:IPV &lt;wsp&gt; &lt;Criterion&gt;, &lt;Next Header&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Next Header:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the IPv6 Next Header.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:NHE:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:NHE:IPV? 1</pre> <p>Returns: 23</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLAB:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLAB:IPV?</pre>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:NHEader:IPV?

<b>Description</b>	<p>This query returns the value of IPv6 Next Header, if the filter type is IPv6 Next Header for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type&gt; Field Match &gt; CONFIg &gt; Filter CONFIg</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:NHEader:IPV? <wsp> <Criterion>[, <Next Header>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Next Header:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the IPv6 Next Header.</p> <p>This parameter is optional. If no token is specified, the current Next Header value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Precedence>
<b>Response(s)</b>	<p><b>Precedence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of IPv6 Next Header.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:NHE:IPV 1, 23</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:NHE:IPV? 1</p> <p>Returns: 23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLAB:IPV</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:FLAB:IPV?</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:FLABel:IPV**

<b>Description</b>	<p>This command sets the IPv6 Flow Label, if the filter type is IPv6 Flow Label for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:FLABel:IPV <wsp> <Criterion>, <Next Header>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Next Header:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the IPv6 Flow Label.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:FLAB:IPV 1, 23</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:FLAB:IPV? 1</p> <p>Returns: 23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHE:IPV</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHE:IPV?</p>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:FLABel:IPV?**

<b>Description</b>	<p>This query returns the value of IPv6 Flow Label, if the filter type is IPv6 Flow Label for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:FLABel:IPV? &lt;wsp&gt; &lt;Criterion&gt;[, &lt;Next Header&gt;   MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Next Header:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the IPv6 Flow Label.</p> <p>This parameter is optional. If no token is specified, the current Flow Label value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Precedence&gt;</p>
<b>Response(s)</b>	<p><b>Precedence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of IPv6 Flow Label.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:FLAB:IPV 1, 23</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:FLAB:IPV? 1</p> <p>Returns: 23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHE:IPV</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:NHE:IPV?</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IPV**

<b>Description</b>	<p>This command sets the source IPv6 address, if the filter type is IPv6 Address Source for specific Filter Number.</p> <p>At *RST condition, this value is set to 0000:0000:0000:0000:0000:0000:0000:0000.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IPV <wsp> <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Source IPv6 address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:IPV 1, "0000:0000:0000:0000:0000:0000:0000:0000" SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:IPV? 1 Returns: "0000:0000:0000:0000:0000:0000:0000:0000"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DEST:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DEST:IPV?</pre>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IPV?

<b>Description</b>	<p>This query returns the Source IPv6 address, if the filter type is IPv6 Address Source for specific Filter Number.</p> <p>At *RST condition, this value is set to 0000:0000:0000:0000:0000:0000:0000:0000.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFIg &gt; Filter CONFIg</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IPV? <wsp> <Criterion>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns source IPv6 address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:IPV 1, "0000:0000:0000:0000:0000:0000:0000:0000"</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:IPV? 1</p> <p>Returns: "0000:0000:0000:0000:0000:0000:0000:0000"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:DEST:IPV</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:DEST:IPV?</p>



**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS:IPV**

<b>Description</b>	<p>This command sets the IPv6 Type of Service (TOS) value, if the filter type is IPv6TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS:IPV <wsp> <Criterion>, <Tos>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the IPv6 TOS value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:TOS:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:TOS:IPV? 1</pre> <p>Returns: 23</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?</pre>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS:IPV?

---

<b>Description</b>	<p>This query returns the IPv6 Type of Service (TOS) value, if the filter type is IPv6TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS:IPV? <wsp> <Criterion>[, <Tos>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the IPv6 TOS value.</p> <p>This parameter is optional. If no token is specified, the current IPv6 TOS value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Tos>
<b>Response(s)</b>	<p><b>Tos:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the IPv6 Type of Service (TOS) value.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:TOS:IPV 1, 23</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:TOS:IPV? 1</p> <p>Returns: 23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:VLAN:ID</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:VLAN:ID?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IPV**

<b>Description</b>	<p>This command sets the Mask Destination IPv6 address, if the filter type is IPv6 Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IPV &lt;wsp&gt;&lt;Criterion&gt;, &lt;Address&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Mask Destination IPv6 address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:IPV 1, "0000:0000:0000:0000:0000:0000:0000:0000" SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:IPV? 1 Returns: "0000:0000:0000:0000:0000:0000:0000:0000"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOUR:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOUR:IPV?</pre>

---

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IPV?**

<b>Description</b>	<p>This query returns the Mask Destination IPv6 address, if the filter type is IPv6 Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IPV? &lt;wsp&gt; &lt;Criterion&gt;</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<p>&lt;Address&gt;</p>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the masked Destination IPv6 address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:IPV 1, "0000:0000:0000:0000:0000:0000:0000:0000"</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:IPV? 1 Returns: "0000:0000:0000:0000:0000:0000:0000:0000"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOUR:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOUR:IPV?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERvices:IPV**

<b>Description</b>	<p>This command sets the Mask IPv6 Differentiated Services, if the filter type is IPv6DiffServ for a specific Filter Number.</p> <p>At *RST condition, this value is set to 63.</p> <p>Navigation Path: Test &gt; Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFIg &gt; Filter CONFIg</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERvices:IPV <wsp> <Criterion>, <Dservices>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask IPv6 Differentiated Services.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, IPVDIFFSERV SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DSER:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DSER:IPV? 1</pre> <p>Returns: 23</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices:IPV?

## SCPI Command Reference

### Filters

---

<b>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERVICES:IPV?</b>	
<b>Description</b>	<p>This query returns the Mask IPv6 Differentiated Services, if the filter type is IPv6DiffServ for a specific Filter Number.</p> <p>At *RST condition, this value is set to 63.</p> <p>Navigation Path: Test &gt; Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERVICES:IPV? &lt;wsp&gt;&lt;Criterion&gt;[, &lt;Dservices&gt;   MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Retrieve the value for masked differentiated services.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Dservices&gt;</code>
<b>Response(s)</b>	<p><b>Dservices:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the masked IPv6 Differentiated Services value.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, IPVDIFFSERV SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DSER:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DSER:IPV? 1 Returns: 23</pre>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:MASK:DSERVICES:IPV</code>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PRECedence:IPV**

<b>Description</b>	<p>This command sets the Mask IPv6 Precedence, if the filter type is IPv6Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Test &gt; Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFIg &gt; Filter CONFIg</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PRECedence:IPV <wsp><Criterion>, <Precedence>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask IPv6 Precedence.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, IPVPRECEDENCE SENS:DATA:TEL:ETH:FMAT:FILT:MASK:PREC:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:PREC:IPV? 1</pre> <p>Returns: 23</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices:IPV?</pre>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PREcedence:IPV?**

<b>Description</b>	<p>This query returns the Mask IPv6 Precedence, if the filter type is IPv6Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Test &gt; Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PREcedence:IPV? &lt;wsp&gt; &lt;Criterion&gt;[, &lt;Precedence&gt;   MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Retrieve the value for masked precedence.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Precedence&gt;</p>
<b>Response(s)</b>	<p><b>Precedence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the masked IPv6 Precedence value.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, IPVPRECEDENCE</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:PREC:IPV 1, 23</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:PREC:IPV? 1</p> <p>Returns: 23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:MASK:DSERvices:IPV</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:MASK:DSERvices:IPV?</p>



**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:NHEader:IPV**

<b>Description</b>	<p>This command sets the Mask IPv6 Next Header, if the filter type is IPv6 Next Header for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:NHEader:IPV &lt;wsp&gt; &lt;Criterion&gt;, &lt;Next Header&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Next Header:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask IPv6 Next Header.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:NHE:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:NHE:IPV? 1</pre> <p>Returns: 23</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer: MASK:FLAB:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer: MASK:FLAB:IPV?</pre>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:NHEader:IPV?**

<b>Description</b>	<p>This query returns the value of Mask IPv6 Next Header, if the filter type is IPv6 Next Header for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:NHEader:IPV? &lt;wsp&gt; &lt;Criterion&gt;[, &lt;Next Header&gt;   MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Next Header:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask IPv6 Next Header.</p> <p>This parameter is optional. If no token is specified, the current Mask IPv6Next Header value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Precedence&gt;</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:NHEader:IPV?**

<b>Response(s)</b>	<b>Precedence:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of Mask IPv6 Next Header.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:FMAT:FILT:MASK:NHE:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:NHE:IPV? 1 Returns: 23
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer: MASK:FLAB:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer: MASK:FLAB:IPV?

---

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:FLABel:IPV**

<b>Description</b>	<p>This command sets the Mask IPv6 Flow Label, if the filter type is IPv6 Flow Label for a specific Filter Number.</p> <p>At *RST condition, this value is set to 1048575.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:FLABel:IPV &lt;wsp&gt; &lt;Criterion&gt;, &lt;Flow Label&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Flow Label:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask IPv6 Flow Label.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:FLAB:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:FLAB:IPV? 1 Returns: 23</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHE:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHE:IPV?</pre>

---

**:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:MASK:FLABel:IPV?**

<b>Description</b>	<p>This query returns the value of Mask IPv6 Flow Label, if the filter type is IPv6 Flow Label for a specific Filter Number.</p> <p>At *RST condition, this value is set to 1048575.</p> <p>Navigation Path: Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:MASK:FLABel:IPV? &lt;wsp&gt; &lt;Criterion&gt;[, &lt;Flow Label&gt;   MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Flow Label:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask IPv6 Flow Label.</p> <p>This parameter is optional. If no token is specified, the current Mask IPv6 Flow Label value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Precedence>

---

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:FLABel:IPV?**

<b>Response(s)</b>	<b>Precedence:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of MaskIPv6 Flow Label.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:FMAT:FILT:MASK:FLAB:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:FLAB:IPV? 1 Returns: 23
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHE:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:NHE:IPV?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IPV**

<b>Description</b>	<p>This command sets the Mask Source IPv6 Address, if the filter type is IPv6 Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IPV <wsp> <Criterion>, <Address>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Mask Source IPv6 address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:IPV 1, "0000:0000:0000:0000:0000:0000:0000:0000" SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:IPV? 1 Returns: "0000:0000:0000:0000:0000:0000:0000:0000"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DEST:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DEST:IPV?</pre>

## SCPI Command Reference

### Filters

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IPV?

---

<b>Description</b>	<p>This query returns the Mask Source IPv6 Address, if the filter type is IPv6 Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IPV? <wsp> <Criterion>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns masked Source IPv6 address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:IPV 1, "0000:0000:0000:0000:0000:0000:0000:0000"</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:IPV? 1 Returns: "0000:0000:0000:0000:0000:0000:0000:0000"</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DEST:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DEST:IPV?

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS:IPV**

<b>Description</b>	<p>This command sets the Mask IPv6 TOS, if the filter type is IPv6 TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Test &gt; Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS:IPV &lt;wsp&gt; &lt;Criterion&gt;, &lt;Tos&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask IPv6 Type of Service (TOS).</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:TOS:IPV 1, 23 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:TOS:IPV? 1</pre> <p>Returns: 23</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS:IPV SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS:IPV?</pre>

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:MASK:TOS:IPV?

<b>Description</b>	<p>This query returns the Mask IPv6 TOS, if the filter type is IPv6 TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.</p> <p>Navigation Path: Test &gt; Functions &gt; Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:MASK:TOS:IPV? <wsp> <Criterion>[, <Tos>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Retrieve the value for masked IPv6 TOS.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Tos>
<b>Response(s)</b>	<p><b>Tos:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the masked IPv6 Type of Service (TOS).</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:TOS:IPV 1, 23</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:TOS:IPV? 1</p> <p>Returns: 23</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:TOS:IPV</p> <p>SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:TOS:IPV?</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MLABel[1..n]**

<b>Description</b>	<p>This command sets the Multi-Protocol Label switching (MPLS), if Filter Type is MPLS Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 1048575.</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFIg &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MLABel[1..n] <wsp><Filterno>, <Criterion>, <Flabel>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Flabel:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the MPLS Label.</p> <p>Range: 0-1048575</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, MLABEL1 SENS:DATA:TEL:ETH:STR:FILT:MLAB1 1, 1, 20 SENS:DATA:TEL:ETH:STR:FILT:MLAB1? 1, 1 Returns: 20</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?</pre>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MLABel[1..n]?**

<b>Description</b>	<p>This query returns the Multi-Protocol Label switching (MPLS), if Filter Type is MPLS Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 1048575.</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MLABel[1..n]? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current MPLS Label value is returned.</p>
<b>Response Syntax</b>	<p>&lt;FLABEL&gt;</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MLABel[1..n]?**

<b>Response(s)</b>	<b>FLABEL:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the MPLS Label.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, MLABEL1 SENS:DATA:TEL:ETH:STR:FILT:MLAB1 1, 1, 20 SENS:DATA:TEL:ETH:STR:FILT:MLAB1? 1, 1 Returns: 20
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?

---

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MCOS[1..n]

<b>Description</b>	<p>This command sets the Multi-Protocol Label switching Cost of Service (MPLS COS), if Filter Type is MPLS COS Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MCOS[1..n] &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;, &lt;Flabel&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Flabel:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the MPLS COS Label.</p> <p>Range: 0-7</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, MCOS 1 SENS:DATA:TEL:ETH:STR:FILT:MCOS1 1, 1, 20 SENS:DATA:TEL:ETH:STR:FILT:MCOS1? 1, 1 Returns: 20</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:TCP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:TCP?</pre>

**:SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:MCOS[1..n]?**

<b>Description</b>	<p>This query returns the Multi-Protocol Label switching Cost of Service(MPLS COS) Label, if Filter Type is MPLS COS Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFig &gt; Value</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:MCOS[1..n]? <wsp> <Filterno>, <Criterion>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current MPLS COS value is returned.</p>
<b>Response Syntax</b>	<FLABEL>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MCOS[1..n]?**

<b>Response(s)</b>	<b>FLABEL:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the MPLS COS Label.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, MCOS1 SENS:DATA:TEL:ETH:STR:FILT:MCOS1 1, 1, 20 SENS:DATA:TEL:ETH:STR:FILT:MCOS1? 1, 1 Returns: 20
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:TCP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:TCP?

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel[1..n]**

<b>Description</b>	<p>This command sets the Multi-Protocol Label switching (MPLS) Mask Next Header, if Filter Type is MPLS Next Header for a specific Filter No.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFIg &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel[1..n] <wsp> <Filterno>, <Criterion>, <Nheader>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Nheader:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the MPLS Mask Next Header.</p> <p>Range:1-1048575</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, MLABEL1 SENS:DATA:TEL:ETH:STR:FILT:MASK:MLAB1 1, 1, 56 SENS:DATA:TEL:ETH:STR:FILT:MASK:MLAB1? 1, 1 Returns: 56</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS1? SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</pre>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel[1..n]?**

<b>Description</b>	<p>This query returns the Multi-Protocol Label switching (MPLS) Mask Next Header, if Filter Type is MPLS Next Header for a specific Filter No.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel[1..n]? &lt;wsp&gt; &lt;Filterno&gt;, &lt;Criterion&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current MPLS Label mask value is returned.</p>
<b>Response Syntax</b>	<p>&lt;Nheader&gt;</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel[1..n]?**

<b>Response(s)</b>	<b>Nheader:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the MPLS Mask Next Header.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, MLABEL1 SENS:DATA:TEL:ETH:STR:FILT:MASK:MLAB1 1, 1, 56 SENS:DATA:TEL:ETH:STR:FILT:MASK:MLAB1? 1, 1 Returns: 56
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS1 SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE

---

## SCPI Command Reference

### Filters

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS[1..n]

<b>Description</b>	<p>This command sets the Multi-Protocol Label switching Cost of Service MPLS COS value of IPv6.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS[1..n] <wsp> <Filterno>, <Criterion>, <Tclass>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tclass:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the MPLS COS value of IPv6 value.</p> <p>Range: 1-7</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, MCOS 1 SENS:DATA:TEL:ETH:STR:FILT:MASK:MCOS1 1, 1, 6 SENS:DATA:TEL:ETH:STR:FILT:MASK:MCOS1? 1, 1 Returns: 6</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel? SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</pre>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS[1..n]?**

<b>Description</b>	<p>This query returns the Multi-Protocol Label switching Cost of Service MPLS COS value of IPv6.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path: Test &gt;Functions &gt;Filters &gt; Filter CONFig &gt; MASK</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS[1..n]? <wsp> <Filterno>, <Criterion>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Filterno:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter number from 1 to 10.</p> <p><b>Criterion:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current MPLS COS mask value is returned.</p>
<b>Response Syntax</b>	<Precedence>

## SCPI Command Reference

### Filters

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS[1..n]?**

<b>Response(s)</b>	<b>Precedence:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns MPLS COS value.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:STR:FILT:TYPE 1, 1, MCOS1 SENS:DATA:TEL:ETH:STR:FILT:MASK:MCOS1 1, 1, 6 SENS:DATA:TEL:ETH:STR:FILT:MASK:MCOS1? 1, 1 Returns: 6
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE

---

## Packet Capture

---

:SOURce[1..n]:DATA:TELEcom:CAPTure:FILTer:TYPE

---

<b>Description</b>	<p>This command sets the capture source for capture configuration.</p> <p>At *RST condition, this value is set to Interface.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Capture Source</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CAPTure:FILTer:TYPE <wsp> <Type>
<b>Parameter(s)</b>	<p><b>Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the capture source type. Allows you to select the filter number 1 to 10. 0 is for Interface selection.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:FILT:TYPE 1</p> <p>SOUR:DATA:TEL:CAPT:FILT:TYPE?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical

---

## SCPI Command Reference

### Packet Capture

---

**:SOURce[1..n]:DATA:TELEcom:CAPTURE:FILTer:TYPE?**

<b>Description</b>	<p>This query returns the capture source for capture configuration.</p> <p>At *RST condition, this value is set to Interface.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Capture Source</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CAPTURE:FILTer:TYPE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the capture source type. 0 to 10 filters.</p> <p>0, interface is selected as the capture source type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:FILT:TYPE 1</p> <p>SOUR:DATA:TEL:CAPT:FILT:TYPE?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical</p> <p>SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical?</p>

---



---

**:SOURce[1..n]:DATA:TELEcom:CAPTure:FRAMe:SIZE**

---

<b>Description</b>	<p>This command sets the frame size type for capture configuration.</p> <p>At *RST condition, this value is set to Complete.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Frame Length</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CAPTure:FRAMe:SIZE <wsp>COMplete   TRUNCATE
<b>Parameter(s)</b>	<p><b>Size Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: COMplete   TRUNCATE</p> <p>Selects the Frame Size type.</p> <p>COMplete: Complete</p> <p>TRUNCATE: Truncated</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:FRAM:SIZE COM</p> <p>SOUR:DATA:TEL:CAPT:FRAM:SIZE?</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:CAPTure:FILTer:TYPE

---

## SCPI Command Reference

### Packet Capture

---

---

#### :SOURce[1..n]:DATA:TELEcom:CAPTure:FRAMe:SIZE?

---

<b>Description</b>	<p>This query returns the frame size type for capture configuration.</p> <p>At *RST condition, this value is set to Complete.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Frame Length</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CAPTure:FRAMe:SIZE?
<b>Response Syntax</b>	<Size Type>
<b>Response(s)</b>	<p><b>Size Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Frame Size type.</p> <p>Complete, COMplete is selected as the Frame Size type.</p> <p>TRUNCATE, TRUncated is selected as Frame Size type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:FRAM:SIZE?</p> <p>SOUR:DATA:TEL:CAPT:FRAM:SIZE COM</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:CAPTure:FILTer:TYPE

---

---

**:SOURce[1..n]:DATA:TELEcom:CAPTure:BYTE**

---

<b>Description</b>	<p>This command sets the number of bytes to capture for capture configuration.</p> <p>At *RST condition, this value is set to 14.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Frame Length &gt; Truncated &gt; Bytes</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CAPTure:BYTE <wsp> <Size>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Size:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value for number of bytes to capture.</p> <p>The valid values are 14 to 1536.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:BYTE 16</p> <p>SOUR:DATA:TEL:CAPT:BYTE?</p> <p>Returns: 16</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:CAPTure:FRAME:SIZE</p> <p>SOURce[1..n]:DATA:TELEcom:CAPTure:FRAME:SIZE?</p>

---

## SCPI Command Reference

### Packet Capture

---

**:SOURce[1..n]:DATA:TELEcom:CAPTure:BYTE?**

<b>Description</b>	<p>This query returns the number of bytes to capture for capture configuration.</p> <p>At *RST condition, this value is set to 14.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Frame Length &gt; Truncated &gt; Bytes</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:CAPTure:BYTE?[ &lt;wsp&gt;MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Set the number of bytes to capture.</p> <p>This parameter is optional. If no token is specified, the current number of bytes to capture is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Byte&gt;</p>
<b>Response(s)</b>	<p><b>Byte:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value for number of bytes to capture.</p> <p>The valid values are 14 to 1536.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:BYTE 16</p> <p>SOUR:DATA:TEL:CAPT:BYTE?</p> <p>Returns: 16</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:CAPTure:FRAME:SIZE?</p>

---

**:SOURCE[1..n]:DATA:TELEcom:CAPTURE:TRIGGER**

---

<b>Description</b>	<p>This command sets the trigger position.</p> <p>At *RST condition, this value is set to Post-trigger.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Trigger &gt; Trigger Position</p>
<b>Syntax</b>	:SOURCE[1..n]:DATA:TELEcom:CAPTURE:TRIGGER <wsp>POST   MID   PRE
<b>Parameter(s)</b>	<p><b>Position:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: POST   MID   PRE</p> <p>Sets the trigger position.</p> <p>PRE: Pre-trigger.</p> <p>MID: Mid-trigger.</p> <p>POST: Post-trigger.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAP:TRIG PRE</p> <p>SOUR:DATA:TEL:CAP:TRIG?</p> <p>Returns: PRE</p>
<b>See Also</b>	SOURCE[1..n]:DATA:TELEcom:CAPTURE:BYTE

---

## SCPI Command Reference

### Packet Capture

---

**:SOURce[1..n]:DATA:TELEcom:CAPTure:TRIGger?**

<b>Description</b>	<p>This query returns the trigger position.</p> <p>At *RST condition, this value is set to Post-trigger.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Trigger &gt; Trigger Position</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CAPTure:TRIGger?
<b>Response Syntax</b>	<Position>
<b>Response(s)</b>	<p><b>Position:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the trigger position.</p> <p>PRE, Pre-trigger is selected as the trigger position.</p> <p>MID, Mid-trigger is selected as the trigger position.</p> <p>POST, Post-trigger is selected as the trigger position.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:TRIG PRE</p> <p>SOUR:DATA:TEL:CAPT:TRIG?</p> <p>Returns: PRE</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:CAPTure:BYTE?

---

---

**:SOURce[1..n]:DATA:TELEcom:CAPTure:TSource**

---

<b>Description</b>	<p>This command sets the Trigger Type.</p> <p>At *RST condition, this value is set to Manual.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Trigger &gt; Trigger Type</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CAPTure:TSource <wsp>ONERROR   FMAatch   MANual
<b>Parameter(s)</b>	<p><b>TYPE:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: ONERROR   FMAatch   MANual</p> <p>Sets the trigger type.</p> <p>ONERROR: ONERROR</p> <p>FMAatch: FMAatch</p> <p>MANual: MANual</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:TSource ONERROR</p> <p>SOUR:DATA:TEL:CAPT:TSource?</p> <p>Returns: ONERROR</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:CAPTure:TRIGger

---

## SCPI Command Reference

### Packet Capture

---

**:SOURce[1..n]:DATA:TELEcom:CAPTURE:TSource?**

<b>Description</b>	<p>This query returns the Trigger Type.</p> <p>At *RST condition, this value is set to Manual.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Trigger &gt; Trigger Type</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CAPTURE:TSource?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the trigger type.</p> <p>ONERROR, returns the trigger type as ONERROR.</p> <p>FMAatch, returns the trigger type as FMAatch.</p> <p>MANual, returns the trigger type as MANual.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:TSource ONERROR</p> <p>SOUR:DATA:TEL:CAPT:TSource?</p> <p>Returns: ONERROR</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:CAPTURE:TRIGger?

---



---

**:SOURce[1..n]:DATA:TELEcom:CAPTure:TSource:TYPE**

<b>Description</b>	<p>This command sets the type of ON Error.</p> <p>At *RST condition, this value is set to FCS.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Trigger &gt; Trigger Type &gt; ON Error</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:CAPTure:TSource:TYPE &lt;wsp&gt;ANYTYPE   IPchecksum   FCS   JABber   RUNt   UNDERSIZED   TCPCHECKSUM   UDPCHECKSUM   OVERSIZED</p>
<b>Parameter(s)</b>	<p><b>TYPE:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: ANYTYPE   IPchecksum   FCS   JABber   RUNt   UNDERSIZED   TCPCHECKSUM   UDPCHECKSUM   OVERSIZED</p> <p>Sets the type of ON error.</p> <p>ANYTYPE: the ANY error trigger source type.</p> <p>IPchecksum: the IP Checksum error trigger source type.</p> <p>FCS: the FCS error trigger source type.</p> <p>JABber: the Jabber error trigger source type.</p> <p>RUNt: the Runt error trigger source type.</p> <p>UNDERSIZED: the Undersized error trigger source type.</p> <p>TCPCHECKSUM: the TCP checksum error trigger source type.</p> <p>UDPCHECKSUM: the UDP checksum error trigger source type.</p> <p>OVERSIZED: the Oversized error trigger source type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:TSource:TYPE FCS</p> <p>SOUR:DATA:TEL:CAPT:TSource:TYPE?</p> <p>Returns: FCS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:CAPTure:TSource</p>

---

## SCPI Command Reference

### Packet Capture

---

**:SOURce[1..n]:DATA:TELEcom:CAPTure:TSource:TYPE?**

<b>Description</b>	<p>This query returns the type of ON Error.</p> <p>At *RST condition, this value is set to FCS.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Trigger &gt; Trigger Type &gt; ON Error</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:CAPTure:TSource:TYPE?
<b>Response Syntax</b>	<Type>
<b>Response(s)</b>	<p><b>Type:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of ON error.</p> <p>ANYTYPE, ANYTYPE error is selected as trigger source type.</p> <p>IPchecksum, IP Checksum error is selected as trigger source type.</p> <p>FCS, FCS error is selected as trigger source type.</p> <p>JABber, Jabber error is selected as trigger source type.</p> <p>RUNt, Runt error is selected as trigger source type.</p> <p>UNDERSIZED, Undersized error is selected as trigger source type.</p> <p>TCPCHECKSUM, TCP checksum error is selected as trigger source type.</p> <p>UDPCHECKSUM, UDP checksum error is selected as trigger source type.</p> <p>OVERSIZED, Oversized error is selected as trigger source type.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:CAPT:TSource:TYPE FCS</p> <p>SOUR:DATA:TEL:CAPT:TSource:TYPE?</p> <p>Returns: FCS</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:CAPTure:TSource:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:CAPTure:TSource:TYPE</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:GLOBal:CONTRol**

<b>Description</b>	<p>This command enables the Capture.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Capture</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:GLOBal:CONTRol <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the Capture.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:GLOB:CONT ON</p> <p>SOUR:DATA:TEL:ETH:GLOB:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:CAPTure:TSource:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:CAPTure:TSource:TYPE</p>

---

## SCPI Command Reference

### Packet Capture

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:GLOBal:CONTRol?**

<b>Description</b>	<p>This query returns the Capture status.</p> <p>This query is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Capture</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:GLOBal:CONTRol?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the Capture.</p> <p>0, returns the capture status as OFF.</p> <p>1, returns the capture status as ON.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:GLOB:CONT ON</p> <p>SOUR:DATA:TEL:ETH:GLOB:CONT?</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:CAPTur:e:TSource:TYPE?</p> <p>SOURce[1..n]:DATA:TELEcom:CAPTur:e:TSource:TYPE</p>

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:STATus?**

<b>Description</b>	<p>This query returns the capture status.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Capture Status</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:STATus?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the capture status.</p> <p>PENDING, returns the capture status as PENDING.</p> <p>ARMED, returns the capture status as ARMED.</p> <p>CAPTURING, returns the capture status as CAPTURING.</p> <p>COMPLETED, returns the capture status as COMPLETED.</p>
<b>Example(s)</b>	FETCh:DATA:TELEcom:ETHernet:STATus?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:CFG:STATus?

---

## SCPI Command Reference

### Packet Capture

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:BUFFer:UTILization?

---

<b>Description</b>	<p>This query returns Buffer Utilization value in percentage.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Buffer Usage</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:BUFFer:UTILization?
<b>Response Syntax</b>	<Buffer>
<b>Response(s)</b>	<p><b>Buffer:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Buffer Utilization value.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:BUFF:UTIL?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:STATus?

---

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:CFG:STATus?**

---

<b>Description</b>	<p>This query returns the configuration status.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Trigger &gt; CFG Status</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:CFG:STATus?
<b>Response Syntax</b>	<CFG Status>
<b>Response(s)</b>	<p><b>CFG Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the configuration status.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:CFG:STAT?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:BUFFer:UTILization?

---

## SCPI Command Reference

### Packet Capture

---

---

#### :FETCh[1..n]:DATA:TELEcom:ETHernet:FRAMe:COUNT?

---

<b>Description</b>	<p>This query indicates the number of frames captured that match the selected filter criteria.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Frame Count</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:FRAMe:COUNT?
<b>Response Syntax</b>	<FrameCount>
<b>Response(s)</b>	<p><b>FrameCount:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frame count.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:FRAM:COUN?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:CFG:STATus?

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:TRIGger:ERRor?**

<b>Description</b>	<p>This query returns the Trigger error.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Trigger Error</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:TRIGger:ERRor?
<b>Response Syntax</b>	<TriggerError>
<b>Response(s)</b>	<p><b>TriggerError:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the Trigger error.</p> <p>ANYTYPE, ANYTYPE error is selected as trigger source type.</p> <p>IPchecksum, IP Checksum error is selected as trigger source type.</p> <p>FCS, FCS error is selected as trigger source type.</p> <p>JABber, Jabber error is selected as trigger source type.</p> <p>RUNt, Runt error is selected as trigger source type.</p> <p>UNDERSIZED, Undersized error is selected as trigger source type.</p> <p>TCPCHECKSUM, TCP checksum error is selected as trigger source type.</p> <p>UDPCHECKSUM, UDP checksum error is selected as trigger source type.</p> <p>OVERSIZED, Oversized error is selected as trigger source type.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:TRIG:ERR?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:FRAME:COUNt?

---

## Triggered Frame Details

---

:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:FNUMBER?

---

<b>Description</b>	<p>This query displays the trigger frame number.</p> <p>This query is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Details &gt; Trigger Frame Details</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:FNUMBER?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Trigger Source frame number.</p>
<b>Example(s)</b>	SENS:DATA:TEL:CAPT:TSource:FNUM?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:PORT?

---

---

**:SENSe[1..n]:DATA:TELeom:CAPTurE:TSource:SOURce:MAC?**

---

<b>Description</b>	<p>This query displays the trigger source MAC Address.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Details &gt; Trigger Frame Details</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:CAPTurE:TSource:SOURce:MAC?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Trigger source MAC address.</p>
<b>Example(s)</b>	SENS:DATA:TEL:CAPT:TSource:SOUR:MAC?
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:CAPTurE:TSource:SOURce:IP?

---

## SCPI Command Reference

### *Triggered Frame Details*

---

**:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:IP?**

<b>Description</b>	<p>This query displays the trigger source IP address.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Details &gt; Trigger Frame Details</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:IP?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Trigger source IP address.</p>
<b>Example(s)</b>	SENS:DATA:TEL:CAPT:TSource:SOUR:IP?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:MAC?

---

---

**:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:PORT?**

---

<b>Description</b>	<p>This query displays the trigger source PORT number.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Details &gt; Trigger Frame Details</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:SOURce:PORT?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Trigger source port number.</p>
<b>Example(s)</b>	SENS:DATA:TEL:CAPT:TSource:SOUR:PORT?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:FNUMber?

---

## SCPI Command Reference

### *Triggered Frame Details*

---

**:SENSe[1..n]:DATA:TELEcom:CAPTurE:TSource:DESTination:MAC?**

<b>Description</b>	<p>This query displays the trigger Destination MAC address.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Details &gt; Trigger Frame Details</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:CAPTurE:TSource:DESTination:MAC?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Trigger destination MAC address.</p>
<b>Example(s)</b>	SENS:DATA:TEL:CAPT:TSource:DEST:MAC?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:CAPTurE:TSource:DESTination:IP?

---

**:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:IP?**

<b>Description</b>	<p>This query displays the trigger Destination IP address.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Details &gt; Trigger Frame Details</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:IP?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Trigger destination IP address.</p>
<b>Example(s)</b>	SENS:DATA:TEL:CAPT:TSource:DEST:IP?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:MAC?

---

## SCPI Command Reference

### *Triggered Frame Details*

---

**:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:PORT?**

<b>Description</b>	<p>This query displays the trigger Destination Port number.</p> <p>At *RST condition, this value is set to device-dependent.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Status And Controls &gt; Details &gt; Trigger Frame Details</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:DESTination:PORT?
<b>Response Syntax</b>	<Result>
<b>Response(s)</b>	<p><b>Result:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Trigger destination port number.</p>
<b>Example(s)</b>	SENS:DATA:TEL:CAPT:TSource:DEST:PORT?
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:CAPTure:TSource:FNUMber?

---



## Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TYPE**

---

**Description**

This command selects the type of filter.

At \*RST condition, this value is set to None.

Navigation Path: Functions > Filters & Packet Capture > Packet Capture > Field Match > Configuration > Filter Configuration

**Syntax**

:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TYPE <wsp> <Criterion>,  
NONE | DMAC | SMAC | VLANid | VPriority | VLAN2id | V2Priority | VLAN3id |  
V3Priority | IPDestination | IPSource | TOS | PRECedence | DSERvices | UDEStination |  
USource | ETHertype | IPPRotocol | IPVDESTINATION | IPVSOURCE | IPVFLABEL |  
IPVNHEADER | IPVDIFFSERV | IPVPRECEDENCE | IPVTOS | MLABEL1 | MLABEL2 |  
MCOS1 | MCOS2

---

**:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:TYPE**

<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Type:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: NONE   DMAC   SMAC   VLANid   VPriority   VLAN2id   V2Priority   VLAN3id   V3Priority   IPDestination   IPSource   TOS   PREcedence   DSERvices   UDEStination   USOurce   ETHertype   IPPRotocol   IPVDESTINATION   IPVSOURCE   IPVFLABEL   IPVNHEADER   IPVDIFFSERV   IPVPRECEDENCE   IPVTOS   MLABEL1   MLABEL2   MCOS1   MCOS2</p> <p>Selects the type of filter.</p> <p>NONE: No filter; DMAC: Destination MAC address; SMAC: the Source MAC address; VLANid: C-VLAN ID; VPriority: C-VLAN Priority; VLAN2id: S-VLAN ID; V2Priority: S-VLAN Priority; VLAN3id: E-VLAN ID; V3Priority: E-VLAN Priority; IPDestination: the Destination IP address; IPSource: the IP Source address; TOS: Type of Service (TOS); PREcedence: Precedence; DSERvices: Differentiated Services; UDEStination: UDP Destination port; USOurce: UDP Source port; ETHertype: Ether Type; IPPRotocol: IP Protocol; IPVDESTINATION: IPv6 Destination IP address; IPVSOURCE: IPv6 Source address; IPVFLABEL: IPv6 Flow Label; IPVNHEADER: IPv6 Next Header; IPVDIFFSERV: IPv6 Differentiated Services; IPVPRECEDENCE: IPv6 Precedence; IPVTOS: IPv6 Type of Service (TOS); MLABEL1: MPLS Label 1; MLABEL2: MPLS Label 2; MCOS1: MPLS COS 1; MCOS2: MPLS COS 2</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, DMAC</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE? 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:TYPE?</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TYPE?**

<b>Description</b>	<p>This query returns the type of filter.</p> <p>At *RST condition, this value is set to None.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TYPE? <wsp> <Criterion>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<FilterType>
<b>Response(s)</b>	<p><b>FilterType:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the type of filter.</p> <p>NONE: No filter; DMAC: Destination MAC address; SMAC: the Source MAC address; VLANid: C-VLAN ID; VPriority: C-VLAN Priority; VLAN2id: S-VLAN ID; V2Priority: S-VLAN Priority; VLAN3id: E-VLAN ID; V3Priority: E-VLAN Priority; IPDestination: the Destination IP address; IPSource: the IP Source address; TOS: Type of Service (TOS); PRECedence: Precedence; DSERvices: Differentiated Services; UDESTination: UDP Destination port; USOurce: UDP Source port; ETHertype: Ether Type; IPPRotocol: IP Protocol; IPVDESTINATION: IPv6 Destination IP address; IPVSOURCE: IPv6 Source address; IPVFLABEL: IPv6 Flow Label; IPVNHEADER: IPv6 Next Header; IPVDIFFSERV: IPv6 Differentiated Services; IPVPRECEDENCE: IPv6 Precedence; IPVTOS, indicates IPv6 Type of Service (TOS);MLABEL1: MPLS Label 1; MLABEL2: MPLS Label 2; MCOS1: MPLS COS 1; MCOS2: MPLS COS 2</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, DMAC</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE? 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE?</p>

## SCPI Command Reference

### *Filter Configuration*

---

---

#### **:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IP**

---

<b>Description</b>	<p>This command sets the Destination Internet Protocol (IP) address, if the filter type is IP Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.0.0.0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IP &lt;wsp&gt; &lt;Criterion&gt;, &lt;Address&gt;</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Destination IP address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:DEST:IP 2,"11.22.33.44" SENS:DATA:TEL:ETH:FMAT:FILT:DEST:IP? 2 Returns: "11.22.33.44"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC?</pre>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IP?**

---

<b>Description</b>	<p>This query returns the Destination Internet Protocol (IP) address, if the filter type is IP Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.0.0.0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:IP? &lt;wsp&gt; &lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;Address&gt;</code>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns destination IP address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:DEST:IP 2,"11.22.33.44" SENS:DATA:TEL:ETH:FMAT:FILT:DEST:IP? 2 Returns: "11.22.33.44"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:MAC?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:MAC**

<b>Description</b>	<p>This command sets the Destination Media Access Control (MAC) address, if the filter type is MAC Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to 00:00:00:00:00:00.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:MAC &lt;wsp&gt; &lt;Criterion&gt;, &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Destination MAC address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:DEST:MAC 2,"00:00:00:FF:FF:FF" SENS:DATA:TEL:ETH:FMAT:FILT:DEST:MAC? 2 Returns: "00:00:00:FF:FF:FF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:MAC?**

<b>Description</b>	<p>This query returns the Destination Media Access Control (MAC) address, if the filter type is MAC Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to 00:00:00:00:00:00.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:MAC? <wsp><Criterion>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns destination MAC address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:DEST:MAC 2,"00:00:00:FF:FF:FF"</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:DEST:MAC? 2</p> <p>Returns: "00:00:00:FF:FF:FF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:IP?</p>

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:UDP**

<b>Description</b>	<p>This command sets the User Datagram Protocol (UDP) Destination Port, if filter type is UDP Destination Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:UDP &lt;wsp&gt; &lt;Criterion&gt;, &lt;port&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>port:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Destination UDP port address.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:DEST:UDP 2,50</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:DEST:UDP? 2</p> <p>Returns: 50</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?</p>

---



**:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:DESTination:UDP?**

<b>Description</b>	<p>This query returns the User Datagram Protocol (UDP) Destination Port, if the filter type is UDP Destination Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:DESTination:UDP? <wsp> <Criterion>[, <Port>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Port:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Destination UDP port address.</p> <p>This parameter is optional. If no token is specified, the current Destination UDP port value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<port>

## SCPI Command Reference

### *Filter Configuration*

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:UDP?**

<b>Response(s)</b>	<b>port:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns destination UDP port address.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:FMAT:FILT:DEST:UDP 2,50 SENS:DATA:TEL:ETH:FMAT:FILT:DEST:UDP? 2 Returns: 50
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERVICES**

<b>Description</b>	<p>This command sets the Differentiated Services, if the filter type is Diff Serv for a specific Filter Number.</p> <p>At *RST condition, this value is set to 000000.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERVICES <wsp> <Criterion>, <Dservices>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Differentiated Services.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:DSER 2,55</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:DSER? 2</p> <p>Returns: 55</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence?</p>

## SCPI Command Reference

### Filter Configuration

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERVICES?

---

<b>Description</b>	<p>This query returns the Differentiated Services, if the filter type is Diff Serv for a specific Filter Number.</p> <p>At *RST condition, this value is set to 000000.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DSERVICES? &lt;wsp&gt; &lt;Criterion&gt;[, &lt;Dservices&gt;   MAXimum   MINimum]</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Differentiated Services.</p> <p>This parameter is optional. If no token is specified, the current Dservices value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<pre>&lt;Dservices&gt;</pre>
<b>Response(s)</b>	<p><b>Dservices:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of differentiated services.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:DSER 2,55 SENS:DATA:TEL:ETH:FMAT:FILT:DSER? 2 Returns: 55</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:PRECEdence?</pre>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECEdence**

---

<b>Description</b>	<p>This command sets the Precedence, if the filter type is Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to 000.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECEdence &lt;wsp&gt; &lt;Criterion&gt;, &lt;Precedence&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Precedence.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:PREC 2,4 SENS:DATA:TEL:ETH:FMAT:FILT:PREC? 2</pre> <p>Returns: 4</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECedence?

---

<b>Description</b>	<p>This query returns the value of Precedence, if the filter type is Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to 000.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:PRECedence? &lt;wsp&gt; &lt;Criterion&gt;[, &lt;Precedence&gt;   MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Precedence.</p> <p>This parameter is optional. If no token is specified, the current precedence value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Precedence&gt;</code>
<b>Response(s)</b>	<p><b>Precedence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of precedence.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:PREC 2,4 SENS:DATA:TEL:ETH:FMAT:FILT:PREC? 2 Returns: 4</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DSERvices?</pre>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IP**

<b>Description</b>	<p>This command sets the source IP address, if the filter type is IP Address Source for specific Filter Number.</p> <p>At *RST condition, this value is set to 0.0.0.0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IP &lt;wsp&gt; &lt;Criterion&gt;, &lt;Address&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Source IP address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:IP 2,"0.1.1.1" SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:IP? 2 Returns: "0.1.1.1"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IP?

---

**Description** This query returns the Source IP address, if the filter type is IP Address Source for specific Filter Number.

At \*RST condition, this value is set to 0.0.0.0.

Navigation Path: Functions > Filters & Packet Capture > Packet Capture > Field Match > Configuration > Filter Configuration

**Syntax** :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:IP? <wsp> <Criterion>

**Parameter(s)** **Criterion:**  
The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.  
Selects the filter criterion number from 1 to 4.

**Response Syntax** <Address>

**Response(s)** **Address:**  
The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element.  
Returns source IP address.

**Example(s)** SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:IP 2,"0.1.1.1"  
SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:IP? 2  
Returns: "0.1.1.1"

**See Also** SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC  
SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:MAC?

---



---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:MAC**

---

<b>Description</b>	<p>This command sets the Source MAC address, if the filter type is MAC Address Source for specific Filter Number.</p> <p>At *RST condition, this value is set to 00:00:00:00:00:00.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:MAC &lt;wsp&gt; &lt;Criterion&gt;, &lt;Address&gt;</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Source MAC address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:MAC 2,"00:00:00:FF:FF:FF" SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:MAC? 2 Returns: "00:00:00:FF:FF:FF"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP?</pre>

---

## SCPI Command Reference

### *Filter Configuration*

---

---

#### **:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:MAC?**

---

<b>Description</b>	<p>This query returns the Source MAC address, if the filter type is MAC address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to 00:00:00:00:00:00.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:MAC? &lt;wsp&gt; &lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;Address&gt;</code>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns source MAC address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:MAC 2,"00:00:00:FF:FF:FF" SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:MAC? 2 Returns: "00:00:00:FF:FF:FF"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:SOURce:IP?</pre>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:UDP**

---

<b>Description</b>	<p>This command sets the User Data Protocol (UDP) Source Port, if the filter type is UDP Source Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:UDP &lt;wsp&gt; &lt;Criterion&gt;, &lt;port&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>port:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Source UDP Port address.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:UDP 2,65 SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:UDP? 2</pre> <p>Returns: 65</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:UDP?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:UDP?

<b>Description</b>	<p>This query returns the User Data Protocol (UDP) source port, if the Filter type is UDP source port for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:SOURce:UDP? &lt;wsp&gt; &lt;Criterion&gt;[, MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Source UDP Port address.</p> <p>This parameter is optional. If no token is specified, the current Source UDP Port value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;port&gt;</code>
<b>Response(s)</b>	<p><b>port:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns source UDP port address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:UDP 2,65</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:SOUR:UDP? 2</p> <p>Returns: 65</p>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:DESTination:UDP?</code>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS**

<b>Description</b>	<p>This command sets the Type of Service (TOS) value, if the filter type is TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0000 0000.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:TOS <wsp> <Criterion>, <Tos>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the TOS value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:TOS 2,30 SENS:DATA:TEL:ETH:FMAT:FILT:TOS? 2</pre> <p>Returns: 30</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?</pre>

## SCPI Command Reference

### Filter Configuration

---

#### :SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:TOS?

<b>Description</b>	<p>This query returns the Type of Service (TOS) value, if the filter type is TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0000 0000.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:TOS? &lt;wsp&gt; &lt;Criterion&gt;[, &lt;Tos&gt;   MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the TOS value.</p> <p>This parameter is optional. If no token is specified, the current TOS value is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Tos&gt;</code>
<b>Response(s)</b>	<p><b>Tos:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Type of Service (TOS) value.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:TOS 2,30 SENS:DATA:TEL:ETH:FMAT:FILT:TOS? 2 Returns: 30</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:VLAN:ID SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:VLAN:ID?</pre>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:ID**

<b>Description</b>	<p>This command sets the value of VLAN identifier, if the filter type is VLAN ID for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:ID <wsp> <Criterion>, <Vlan>, <Id>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p><b>Id:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value of VLAN Identifier.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:VLAN:ID 2,1, 4095</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:VLAN:ID? 2,1</p> <p>Returns: 4095</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRIority</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRIority?</p>

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:ID?**

<b>Description</b>	<p>This query returns the value of VLAN ID, if the filter type is VLAN ID for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:ID? &lt;wsp&gt; &lt;Criterion&gt;, &lt;Vlan&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the VLAN ID.</p> <p>This parameter is optional. If no token is specified, the current VLAN ID is returned.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;VlanId&gt;</p>

---



---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:ID?**

<b>Response(s)</b>	<b>VlanId:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of VLAN identifier.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:FMAT:FILT:VLAN:ID 2,1, 4095 SENS:DATA:TEL:ETH:FMAT:FILT:VLAN:ID? 2,1 Returns: 4095
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRIOriety SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRIOriety?

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:PRIority**

<b>Description</b>	<p>This command sets the value of VLAN priority, if the filter type is VLAN priority for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:PRIority &lt;wsp&gt; &lt;Criterion&gt;, &lt;Vlan&gt;, &lt;Priority&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p><b>Priority:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value of VLAN Priority.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:VLAN:PRI 2,1,7</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:VLAN:PRI? 2,1</p> <p>Returns: 7</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:PRIority?**

<b>Description</b>	<p>This query returns the value of Virtual Local Area Network (VLAN) priority, if the filter type is VLAN priority for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:PRIority? &lt;wsp&gt; &lt;Criterion&gt;, &lt;Vlan&gt;[, MAXimum   MINimum]</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the VLAN Priority value.</p> <p>This parameter is optional. If no token is specified, the current VLAN Priority value is returned.</p> <p>MAXimum: Biggest supported value MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<code>&lt;Priority&gt;</code>

---

## SCPI Command Reference

### *Filter Configuration*

---

#### **:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:VLAN:PRiority?**

<b>Response(s)</b>	<b>Priority:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the value of VLAN priority.
<b>Example(s)</b>	SENS:DATA:TEL:ETH:FMAT:FILT:VLAN:PRI 2,1, 7 SENS:DATA:TEL:ETH:FMAT:FILT:VLAN:PRI? 2,1 Returns: 7
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:IPPRotocol**

---

<b>Description</b>	<p>This command sets IP Protocol value, if the filter type is IP Protocol for a specific Filter Number.</p> <p>At *RST condition, this value is set to 17.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:IPPRotocol &lt;wsp&gt; &lt;Criterion&gt;, &lt;Ipprotocol&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Ipprotocol:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the filter IP protocol value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMATch:FILT:IPPR 1, 17 SENS:DATA:TEL:ETH:FMATch:FILT:IPPR? 1 Returns: 17</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:IPPRotocol?

<b>Description</b>	<p>This query returns the IP Protocol value, if the filter type is IP Protocol for a specific Filter Number.</p> <p>At *RST condition, this value is set to 17.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:IPPRotocol? <wsp> <Criterion>[, <Ipprotocol>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Ipprotocol:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Gets the filter IP protocol value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Ipprotocol>
<b>Response(s)</b>	<p><b>Ipprotocol:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the IP Protocol value.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMATch:FILT:IPPR 1, 17</p> <p>SENS:DATA:TEL:ETH:FMATch:FILT:IPPR? 1</p> <p>Returns: 17</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:ETHertype</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:ETHertype?</p>

---

**:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:ETHertype**

<b>Description</b>	<p>This command sets the Ether Type value, if the filter type is Ether Type for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0x0800.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:ETHertype &lt;wsp&gt; &lt;Criterion&gt;, &lt;EtherType&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>EtherType:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the filter ether type value.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:ETH 1, #HAB SENS:DATA:TEL:ETH:FMAT:FILT:ETH? 1</pre> <p>Returns: 171</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:IPPRotocol SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:IPPRotocol?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:ETHertype?

---

<b>Description</b>	<p>This query returns the Ether Type value, if the filter type is Ether Type for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0x0800.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:ETHertype? <wsp> <Criterion>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<EtherType>
<b>Response(s)</b>	<p><b>EtherType:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the EtherType value.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:ETH 1, #HAB</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:ETH? 1</p> <p>Returns: 171</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol?</p>

---



---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IP**

<b>Description</b>	<p>This command sets the Mask Destination IP address, if the filter type is IP Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.255.255.255.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IP &lt;wsp&gt;&lt;Criterion&gt;, &lt;Address&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Mask Destination IP address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:IP 2,"255.255.255.255" SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:IP? 2 Returns: "255.255.255.255"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IP?**

<b>Description</b>	<p>This query returns the Mask Destination IP address, if the filter type is IP Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.255.255.255.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:IP? &lt;wsp&gt; &lt;Criterion&gt;</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<p>&lt;Address&gt;</p>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the masked Destination IP address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:IP 2,"255.255.255.255"</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:IP? 2</p> <p>Returns: "255.255.255.255"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:MAC?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:MAC**

<b>Description</b>	<p>This command sets the Mask Destination MAC address, if the filter type is MAC Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to FF:FF:FF:FF:FF:FF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:MAC &lt;wsp&gt; &lt;Criterion&gt;, &lt;Address&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Mask Destination MAC address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:MAC 2,"FF:FF:FF:FF:FF:FF" SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:MAC? 2 Returns: "FF:FF:FF:FF:FF:FF"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP?</pre>

---

## SCPI Command Reference

### *Filter Configuration*

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:MAC?**

<b>Description</b>	<p>This query returns the Mask Destination MAC address, if the filter type is MAC Address Destination for a specific Filter Number.</p> <p>At *RST condition, this value is set to FF:FF:FF:FF:FF:FF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:MAC? &lt;wsp&gt; &lt;Criterion&gt;</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<p>&lt;Address&gt;</p>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the masked Destination MAC address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:MAC 2,"FF:FF:FF:FF:FF:FF" SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:MAC? 2 Returns: "FF:FF:FF:FF:FF:FF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:IP?</p>

---

---

**:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:MASK:DESTination:UDP**

<b>Description</b>	<p>This command sets the Mask UDP Destination Port, if the filter type is UDP Destination Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0xFFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:MASK:DESTination:UDP &lt;wsp&gt; &lt;Criterion&gt;, &lt;port&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>port:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Mask Destination UDP port address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:UDP 2,#HAB SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:UDP? 2</pre> <p>Returns: 171</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:MASK:DESTination:UDP SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:MASK:DESTination:UDP?</pre>

---

## SCPI Command Reference

### *Filter Configuration*

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:UDP?**

<b>Description</b>	<p>This query returns the Mask UDP Destination Port, if the filter type is UDP Destination Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0xFFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DESTination:UDP? &lt;wsp&gt; &lt;Criterion&gt;</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<p>&lt;port&gt;</p>
<b>Response(s)</b>	<p><b>port:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the masked Destination UDP Port address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:UDP 2,#HAB</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DEST:UDP? 2</p> <p>Returns: 171</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:UDP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DESTination:UDP?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERVICES**

---

<b>Description</b>	<p>This command sets the Mask Differentiated Services, if the filter type is Diff Serv for a specific Filter Number.</p> <p>At *RST condition, this value is set to 111111.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:DSERVICES &lt;wsp&gt; &lt;Criterion&gt;, &lt;Dservices&gt;   MAXimum   MINimum</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask Differentiated Services.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DSER 2,21 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DSER? 2</pre> <p>Returns: 21</p>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERVICES?</code>

---

## SCPI Command Reference

### Filter Configuration

---

#### :SENSe[1..n]:DATA:TELeom:ETHernet:FMATch:FILTer:MASK:DSERvices?

<b>Description</b>	<p>This query returns the Mask Differentiated Services, if the filter type is Diff Serv for a specific Filter Number.</p> <p>At *RST condition, this value is set to 111111.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELeom:ETHernet:FMATch:FILTer:MASK:DSERvices? <wsp><Criterion>[, <Dservices>   MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Dservices:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Retrieves the masked differentiated services value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<Dservices>
<b>Response(s)</b>	<p><b>Dservices:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the masked Differentiated Services value.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DSER 2,21</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:DSER? 2</p> <p>Returns: 21</p>
<b>See Also</b>	SENSe[1..n]:DATA:TELeom:ETHernet:STReam:FILTer:MASK:DSERvices



**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PREcedence**

<b>Description</b>	<p>This command sets the Mask Precedence, if the filter type is Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to 111.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PREcedence &lt;wsp&gt; &lt;Criterion&gt;, &lt;Precedence&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask Precedence.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:PREC 2,1</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:PREC? 2</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices?</p>

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PRECedence?**

<b>Description</b>	<p>This query returns the Mask Precedence, if the filter type is Precedence for a specific Filter Number.</p> <p>At *RST condition, this value is set to 111.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:PRECedence? &lt;wsp&gt; &lt;Criterion&gt;[, &lt;Precedence&gt;   MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Precedence:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Retrieves the masked precedence value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Precedence&gt;</p>
<b>Response(s)</b>	<p><b>Precedence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the masked Precedence value.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:PREC 2,1</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:PREC? 2</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:DSERvices?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IP**

---

<b>Description</b>	<p>This command sets the Mask Source IP Address, if the filter type is IP Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.255.255.255.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IP &lt;wsp&gt;&lt;Criterion&gt;, &lt;Address&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Mask Source IP address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:IP 2,"255.255.255.255" SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:IP? 2 Returns: "255.255.255.255"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IP?

---

<b>Description</b>	<p>This query returns the Mask Source IP Address, if the filter type is IP Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to 255.255.255.255.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:IP? <wsp> <Criterion>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Address>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns masked Source IP address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:IP 2,"255.255.255.255"</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:IP? 2</p> <p>Returns: "255.255.255.255"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:MAC?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:MAC**

<b>Description</b>	<p>This command sets the Mask Source MAC Address, if the filter type is MAC Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to FF:FF:FF:FF:FF:FF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:MAC &lt;wsp&gt;&lt;Criterion&gt;, &lt;Address&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Address:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Selects the Mask Source MAC address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:MAC 2,"FF:FF:FF:FF:FF:FF" SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:MAC? 2 Returns: "FF:FF:FF:FF:FF:FF"</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:MAC?**

<b>Description</b>	<p>This query returns the Mask Source MAC Address, if the filter type is MAC Address Source for a specific Filter Number.</p> <p>At *RST condition, this value is set to FF:FF:FF:FF:FF:FF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:MAC? &lt;wsp&gt; &lt;Criterion&gt;</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<p>&lt;Address&gt;</p>
<b>Response(s)</b>	<p><b>Address:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns masked Source MAC address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:MAC 2,"FF:FF:FF:FF:FF:FF"</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:MAC? 2</p> <p>Returns: "FF:FF:FF:FF:FF:FF"</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:IP?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:UDP**

---

<b>Description</b>	<p>This command sets the Mask UDP Source Port, if the filter type is UDP Source Port for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0xFFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:UDP &lt;wsp&gt; &lt;Criterion&gt;, &lt;port&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>port:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the Mask Source UDP Port address.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:UDP 2,#HAB SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:UDP? 2</pre> Returns: 171
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:UDP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:UDP?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:UDP?**

<b>Description</b>	<p>The query returns the Mask UDP Source Port, if the filter type is UDP Source Port for specific Filter Number.</p> <p>At *RST condition, this value is set to 0xFFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:SOURce:UDP? &lt;wsp&gt; &lt;Criterion&gt;</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<p>&lt;port&gt;</p>
<b>Response(s)</b>	<p><b>port:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the masked Source UDP Port address.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:UDP 2,#HAB</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:SOUR:UDP? 2</p> <p>Returns: 171</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:UDP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:SOURce:UDP?</p>

---



---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS**

---

<b>Description</b>	<p>This command sets the Mask TOS, if the filter type is TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to 1111 1111.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS &lt;wsp&gt; &lt;Criterion&gt;, &lt;Tos&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the Mask Type of Service (TOS).</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:TOS 2,60 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:TOS? 2</pre> <p>Returns: 60</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS?**

<b>Description</b>	<p>This query returns the Mask TOS, if the filter type is TOS for a specific Filter Number.</p> <p>At *RST condition, this value is set to 1111 1111.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:TOS? &lt;wsp&gt; &lt;Criterion&gt;[, &lt;Tos&gt;   MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tos:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Retrieves the masked TOS value.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Tos&gt;</p>
<b>Response(s)</b>	<p><b>Tos:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the masked Type of Service (TOS).</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:TOS 2,60</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:TOS? 2</p> <p>Returns: 60</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TOS?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:ID**

---

<b>Description</b>	<p>This command sets the Mask VLAN ID, if the filter type is VLAN ID for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0xFFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:ID &lt;wsp&gt; &lt;Criterion&gt;, &lt;Vlan&gt;, &lt;Id&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the masked VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p><b>Id:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the value of masked VLAN Identifier.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:VLAN:ID 2,1, #HAB SENS:DATA:TEL:ETH:FMAT:FILT:MASK:VLAN:ID? 2,1 Returns: 171</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:ID?**

<b>Description</b>	<p>This query returns the Mask VLAN ID, if the filter type is VLAN ID for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0xFFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:ID? &lt;wsp&gt; &lt;Criterion&gt;, &lt;Vlan&gt;</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the masked VLAN.</p> <p>Vlan stack is always 1   2   3</p>
<b>Response Syntax</b>	<p>&lt;VlanId&gt;</p>
<b>Response(s)</b>	<p><b>VlanId:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of masked VLAN identifier.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:VLAN:ID 2,1, #HAB</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:VLAN:ID? 2,1</p> <p>Returns: 171</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:ID?</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:PRiority**

<b>Description</b>	<p>This command sets the Mask VLAN Priority, if the filter type is VLAN Priority for a specific Filter Number.</p> <p>At *RST condition, this value is set to 111.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:PRiority <wsp> <Criterion>, <Vlan>, <Priority>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the masked VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p><b>Priority:</b></p> <p>The program data syntax for the third parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the value of masked VLAN Priority.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:VLAN:PRI 2,1,2 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:VLAN:PRI? 2,1</pre> <p>Returns: 2</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority?</pre>

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:PRIority?**

<b>Description</b>	<p>This query returns the Mask VLAN Priority, if the filter type is VLAN Priority for a specific Filter Number.</p> <p>At *RST condition, this value is set to 111.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:PRIority? &lt;wsp&gt; &lt;Criterion&gt;, &lt;Vlan&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Vlan:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>Selects the masked VLAN.</p> <p>Vlan stack is always 1   2   3</p> <p>The program data syntax for the third parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Selects the masked VLAN Priority.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p>
<b>Response Syntax</b>	<p>&lt;Priority&gt;</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:VLAN:PRiority?**

<b>Response(s)</b>	<p><b>Priority:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the value of masked VLAN Priority.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:VLAN:PRI 2,1, 2</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:VLAN:PRI? 2,1</p> <p>Returns: 2</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:VLAN:PRiority?</p>

---

## SCPI Command Reference

### Filter Configuration

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:IPPRotocol

---

<b>Description</b>	<p>This command sets Mask IP Protocol, if the filter type is IP Protocol for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0xFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:IPPRotocol <wsp> <Criterion>, <Ipprotocol>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Ipprotocol:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Mask IP protocol value.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:IPPR 2,#HFF SENS:DATA:TEL:ETH:FMAT:FILT:MASK:IPPR? 2</pre> Returns: 255
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol?</pre>

---



---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:IPPRotocol?**

---

<b>Description</b>	<p>This query returns the Mask IP Protocol, if the filter type is IP Protocol for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0xFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:IPPRotocol? &lt;wsp&gt;&lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;Ipprotocol&gt;</code>
<b>Response(s)</b>	<p><b>Ipprotocol:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Masked IP Protocol.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:IPPR 2,#HFF SENS:DATA:TEL:ETH:FMAT:FILT:MASK:IPPR? 2 Returns: 255</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:IPPRotocol?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:ETHertype**

<b>Description</b>	<p>This command sets Mask Ether Type, if the filter type is Ether Type for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0xFFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:ETHertype &lt;wsp&gt; &lt;Criterion&gt;, &lt;EtherType&gt;</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>EtherType:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;NONDECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Sets the Mask filter ether type value.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:ETH 2,#HAB</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:ETH? 2</p> <p>Returns: 171</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:ETHertype?**

---

<b>Description</b>	<p>This query returns the Mask Ether Type, if the filter type is Ether Type for a specific Filter Number.</p> <p>At *RST condition, this value is set to 0xFFFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:ETHertype? &lt;wsp&gt; &lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;EtherType&gt;</code>
<b>Response(s)</b>	<p><b>EtherType:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the Masked EtherType value.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:ETH 2,#HAB SENS:DATA:TEL:ETH:FMAT:FILT:MASK:ETH? 2 Returns: 171</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:ETHertype?</pre>

---

## SCPI Command Reference

### *Filter Configuration*

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator

---

<b>Description</b>	<p>This command selects the logical operator (AND or OR) between two operands when more than two operands are used.</p> <p>At *RST condition, this value is set to AND.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration &gt; Oper.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator <wsp> <Criterion>, AND   OR
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Operator:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AND   OR</p> <p>Selects the logical operators.</p> <p>AND</p> <p>OR</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:OPER 1, AND</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:OPER? 1</p> <p>Returns: AND</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator?**

<b>Description</b>	<p>This query returns the logical operator (AND or OR) between two operands when more than two operands are used.</p> <p>At *RST condition, this value is set to AND.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration &gt; Oper.</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator? <wsp> <Criterion>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<Operator>
<b>Response(s)</b>	<p><b>Operator:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the logical operators.</p> <p>AND, AND is returned as the logical operator.</p> <p>OR, OR is returned as the logical operator.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:OPER 2,AND</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:OPER? 2</p> <p>Returns: AND</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:OPERator</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STREam:FILTer:OPERator?</p>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator:NOT**

<b>Description</b>	<p>This command selects the Operator Not. When it is selected, add the logical negation (not equal) operator for the operand filter defined.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration &gt; Not</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator:NOT &lt;wsp&gt; &lt;Criterion&gt;</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>Enables or disables the NOT operator.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:OPER:NOT 2,ON</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:OPER:NOT? 2</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT?</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator:NOT?**

---

<b>Description</b>	<p>This query returns the When selected, add the logical negation (not equal) operator for the operand filter defined at its right.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration &gt; Not</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:OPERator:NOT? &lt;wsp&gt; &lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;Set&gt;</code>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the NOT operator.</p> <p>1, NOT is enabled.</p> <p>0, NOT is disabled.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:OPER:NOT 2,ON SENS:DATA:TEL:ETH:FMAT:FILT:OPER:NOT? 2 Returns: 1</pre>
<b>See Also</b>	<code>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT</code>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:OPEN**

<b>Description</b>	<p>This command selects the open parenthesis to control the precedence of operands when two operands are used.</p> <p>At *RST condition, this value is set to blank.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration (</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:OPEN &lt;wsp&gt; &lt;Criterion&gt;, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the open parenthesis.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:BRAC:OPEN 2,ON</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:BRAC:OPEN? 2</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:CLOSe?</p>



---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:OPEN?**

---

<b>Description</b>	<p>This query returns the selected open parenthesis to control the precedence of operands when two operands are used.</p> <p>At *RST condition, this value is set to blank.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration (</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:OPEN? &lt;wsp&gt;&lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;Set&gt;</code>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of open parenthesis.</p> <p>1, bracket open is enabled.</p> <p>0, bracket open is disabled.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:BRAC:OPEN 2,ON SENS:DATA:TEL:ETH:FMAT:FILT:BRAC:OPEN? 2 Returns: 1</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:OPEN SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:BRACket:OPEN?</pre>

---

## SCPI Command Reference

### *Filter Configuration*

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:CLOSe**

<b>Description</b>	<p>This command selects the close parenthesis to control the precedence of operands when two operands are used.</p> <p>At *RST condition, this value is set to blank.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration )</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:CLOSe &lt;wsp&gt; &lt;Criterion&gt;, ON   OFF</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Set:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enables or disables the close parenthesis.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:BRAC:CLOS 2,ON</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:BRAC:CLOS? 2</p> <p>Returns: 1</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT?</p>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:CLOSe?**

---

<b>Description</b>	<p>This query returns the selected close parenthesis to control the precedence of operands when two operands are used.</p> <p>At *RST condition, this value is set to blank.</p> <p>Navigation Path: Test &gt; Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; Configuration &gt; Filter Configuration )</p>
<b>Syntax</b>	<code>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:BRACket:CLOSe? &lt;wsp&gt; &lt;Criterion&gt;</code>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p>
<b>Response Syntax</b>	<code>&lt;Set&gt;</code>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of close parenthesis.</p> <p>1, returns the close parenthesis is enabled.</p> <p>0, returns the close parenthesis is disabled.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:BRAC:CLOS 2,ON SENS:DATA:TEL:ETH:FMAT:FILT:BRAC:CLOS? 2 Returns: 1</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:OPERator:NOT?</pre>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:MLABel[1..n]**

<b>Description</b>	<p>This command sets the Multi-Protocol Label switching (MPLS), if Filter Type is MPLS Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 1048575.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELecom:ETHernet:FMATch:FILTer:MLABel[1..n] &lt;wsp&gt; &lt;Criterion&gt;, &lt;FLabel&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>FLabel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the MPLS Label.</p> <p>Range:0-1048575</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, MLABEL1 SENS:DATA:TEL:ETH:FMAT:FILT:MLAB1 1, 20 SENS:DATA:TEL:ETH:FMAT:FILT:MLAB1? 1</pre> <p>Returns: 20</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:DESTination:UDP SENSe[1..n]:DATA:TELecom:ETHernet:STReam:FILTer:DESTination:UDP?</pre>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MLABel[1..n]?**

<b>Description</b>	<p>This query returns the Multi-Protocol Label switching (MPLS), if Filter Type is MPLS Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 1048575.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MLABel[1..n]? <wsp> <Criterion>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current MPLS label value is returned.</p>
<b>Response Syntax</b>	<FLABEL>
<b>Response(s)</b>	<p><b>FLABEL:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the MPLS Label.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, MLABEL1</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MLAB1 1, 20</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MLAB1? 1</p> <p>Returns: 20</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:UDP?</p>

## SCPI Command Reference

### Filter Configuration

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MCOS[1..n]

---

<b>Description</b>	<p>This command sets the Multi-Protocol Label switching Cost of Service (MPLS COS), if Filter Type is MPLS COS Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MCOS[1..n] <wsp> <Criterion>, <Flabel>   MAXimum   MINimum
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Flabel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the MPLS COS Label.</p> <p>Range:0-7</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, MCOS1 SENS:DATA:TEL:ETH:FMAT:FILT:MCOS1 1, 6 SENS:DATA:TEL:ETH:FMAT:FILT:MCOS1? 1</pre> <p>Returns: 6</p>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:TCP SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:TCP?</pre>

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MCOS[1..n]?**

<b>Description</b>	<p>This query returns the Multi-Protocol Label switching Cost of Service (MPLS COS) Label, if Filter Type is MPLS COS Label for a specific Filter No.</p> <p>At *RST condition, this value is set to 7.</p> <p>Navigation Path: Functions &gt; Filters &amp; Packet Capture &gt; Packet Capture &gt; Field Match &gt; CONFig &gt; Filter CONFig</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MCOS[1..n]? &lt;wsp&gt; &lt;Criterion&gt;[, MAXimum   MINimum]</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current MPLS COS value is returned.</p>
<b>Response Syntax</b>	<p>&lt;FLABEL&gt;</p>
<b>Response(s)</b>	<p><b>FLABEL:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the MPLS COS Label.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, MCOS1</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MCOS1 1, 6</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MCOS1? 1</p> <p>Returns: 6</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:TCP</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:DESTination:TCP?</p>

---

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MLABel[1..n]**

<b>Description</b>	<p>This command sets the Multi-Protocol Label switching (MPLS) Mask Next Header, if Filter Type is MPLS Next Header for a specific Filter No.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Functions-&gt;Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFIg &gt; Filter CONFIg &gt;Mask</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MLABel[1..n] &lt;wsp&gt; &lt;Criterion&gt;, &lt;Nheader&gt;   MAXimum   MINimum</pre>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Nheader:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the MPLS Mask Next Header.</p> <p>Range:1-1048575</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, MLABEL1 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:MLAB1 1, 56 SENS:DATA:TEL:ETH:FMAT:FILT:MASK:MLAB1? 1 Returns: 56</pre>
<b>See Also</b>	<pre>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS1? SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</pre>

---



---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MLABel[1..n]?**


---

<b>Description</b>	<p>This query returns the Multi-Protocol Label switching (MPLS) Mask Next Header, if Filter Type is MPLS Next Header for a specific Filter No.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Functions-&gt;Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFIg &gt; Filter CONFIg &gt;Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MLABel[1..n]? <wsp><Criterion>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current MPLS mask Label value is returned.</p>
<b>Response Syntax</b>	<Nheader>
<b>Response(s)</b>	<p><b>Nheader:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the MPLS Mask Next Header.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, MLABEL1</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:MLAB1 1, 56</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:MLAB1? 1</p> <p>Returns: 56</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MCOS1</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p> <hr/>

## SCPI Command Reference

### Filter Configuration

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MCOS[1..n]**

<b>Description</b>	<p>This command sets the Multi-Protocol Label switching Cost of Service MPLS COS value of IPv6.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Functions-&gt;Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig &gt;Mask</p>
<b>Syntax</b>	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MCOS[1..n] &lt;wsp&gt; &lt;Criterion&gt;, &lt;Tclass&gt;   MAXimum   MINimum</p>
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p><b>Tclass:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;numeric_value&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>Sets the MPLS COS value of IPv6 value.</p> <p>Range:1-7</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, MCOS1</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:MCOS1 1, 4</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:MCOS1? 1</p> <p>Returns: 4</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel?</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p>

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MCOS[1..n]?**

<b>Description</b>	<p>This query returns the Multi-Protocol Label switching Cost of Service MPLS COS value of IPv6.</p> <p>At *RST condition, this value is set to 1.</p> <p>Navigation Path:Functions-&gt;Packet Capture &gt; Trigger Type &gt; Field Match &gt; CONFig &gt; Filter CONFig &gt;Mask</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:FMATch:FILTer:MASK:MCOS[1..n]? <wsp> <Criterion>[, MAXimum   MINimum]
<b>Parameter(s)</b>	<p><b>Criterion:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the filter criterion number from 1 to 4.</p> <p>The program data syntax for the second parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MAXimum   MINimum</p> <p>This parameter is optional. If no token is specified, the current MPLS Mask COS value is returned.</p>
<b>Response Syntax</b>	<Precedence>
<b>Response(s)</b>	<p><b>Precedence:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns MPLS COS value.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:FMAT:FILT:TYPE 1, MCOS1</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:MCOS1 1, 4</p> <p>SENS:DATA:TEL:ETH:FMAT:FILT:MASK:MCOS1? 1</p> <p>Returns: 4</p>
<b>See Also</b>	<p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:MASK:MLABel</p> <p>SENSe[1..n]:DATA:TELEcom:ETHernet:STReam:FILTer:TYPE</p>

## GMP

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CMStatus?**

<b>Description</b>	<p>This query returns the CM status transmitted during the test.</p> <p>GMP is present for the following OTN Multiplexing rates: ODU3(P<sub>T</sub>=21)/ODU0 or ODU3(P<sub>T</sub>=21)/ODU0 or ODU3(P<sub>T</sub>=21)/ODU2.</p> <p>This query is not associated with any *RST condition.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Functions &gt; GMP</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CMStatus? &lt;wsp&gt;MINValue   MAXValue[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>CMStatus:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MINValue   MAXValue</p> <p>Displays the CM value transmitted.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>

**:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:GMP:TX:CMStatus?**

<b>Response(s)</b>	<b>Value:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the CM status transmitted during the test. MINValue, displays the minimum CM value transmitted. MAXValue, displays the maximum CM value transmitted.
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ODU3:GMP:TX:CMST? MAXV
<b>See Also</b>	FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:GMP:TX:CNDSStatus?

---

## SCPI Command Reference

### GMP

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CNdStatus?

<b>Description</b>	<p>This query returns the CNd status transmitted during the test.</p> <p>GMP is present for the following OTN Multiplexing rates: ODU3(Pt=21)/ODU0 or ODU3(Pt=21)/ODU0 or ODU3(Pt=21)/ODU2.</p> <p>This query is not associated with any *RST condition.</p> <p>Navigation Path: Test &gt; OTN BERT &gt; Functions &gt; GMP</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CNdStatus? <wsp>MINValue   MAXValue[, <Channel>]
<b>Parameter(s)</b>	<p><b>CNdStatus:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MINValue   MAXValue</p> <p>Displays the CNd value transmitted.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the CNd status transmitted during the test.</p> <p>MINValue, displays the minimum CNd value transmitted.</p> <p>MAXValue, displays the maximum CNd value transmitted.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ODU3:GMP:TX:CNdS? MINV
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CMStatus?

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:CMStatus?**

<b>Description</b>	<p>This query returns the CM status captured during the test.</p> <p>This query is not associated with any *RST condition.</p> <p>Navigation Path: OTN BERT &gt; Functions &gt; GMP</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:CMStatus? <wsp>MINValue   MAXValue[, <Channel>]
<b>Parameter(s)</b>	<p><b>CMStatus:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MINValue   MAXValue</p> <p>Displays the CM value captured.</p> <p>MAXimum: Biggest supported value</p> <p>MINimum: Smallest supported value</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the CM status captured during the test.</p> <p>MINValue, displays the minimum CM value captured.</p> <p>MAXValue, displays the maximum CM value captured.</p>
<b>Example(s)</b>	FETC:DATA:TEL:OTN:ODU3:GMP:RX:CMST? MINV
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CMStatus?

## SCPI Command Reference

### GMP

---

---

#### :FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:CNdStatus?

---

<b>Description</b>	<p>This query returns the CNd status captured during the test.</p> <p>This query is not associated with any *RST condition.</p> <p>Navigation Path: OTN BERT &gt; Functions &gt; GMP</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:RX:CNdStatus? &lt;wsp&gt;MINValue   MAXValue[, &lt;Channel&gt;]</p>
<b>Parameter(s)</b>	<p><b>CNdStatus:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: MINValue   MAXValue</p> <p>Sets the CNd status captured during the test.</p> <p>MINValue, displays the minimum CNd value captured.</p> <p>MAXValue, displays the maximum CNd value captured.</p> <p><b>Channel:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Selects the channel that will be used by command/query.</p> <p>This parameter can only be used in Multi-Channel OTN.</p> <p>The numeric channel ranges from [1:n] in function of ODU Mapping.</p>
<b>Response Syntax</b>	<p>&lt;Value&gt;</p>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the CNd status captured during the test.</p> <p>MINValue, displays the minimum CNd value captured.</p> <p>MAXValue, displays the maximum CNd value captured.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:OTN:ODU3:GMP:RX:CNDS? MINV</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:GMP:TX:CNdStatus?</p>

---



## Client Offset

---

**:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency?**

---

<b>Description</b>	<p>This query returns the frequency used for transmission.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Tx Frequency &gt; Frequency</p> <p>Navigation Path for IGBE client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Tx Frequency &gt; Frequency</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency? <wsp>AFRequency   NFRequency
<b>Parameter(s)</b>	<p><b>Frequency Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: AFRequency   NFRequency</p> <p>Indicates the frequency used for transmission.</p> <p>AFRequency, Indicates the frequency (Nominal frequency + port frequency offset + client frequency offset) used for transmission of the client signal.</p> <p>NFRequency, Indicates the nominal frequency of the signal.</p>
<b>Response Syntax</b>	<Frequency>
<b>Response(s)</b>	<p><b>Frequency:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the frequency value used for transmission.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:COFF:FREQ? AFR
<b>See Also</b>	SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency?

---

## SCPI Command Reference

### *Client Offset*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:OFFSet**

<b>Description</b>	<p>This command allows entering a positive or a negative client frequency offset in ppm.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Tx Frequency &gt; Offset value</p> <p>Navigation Path for 1GbE client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Tx Frequency &gt; Offset</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:OFFSet <wsp> <CFO>
<b>Parameter(s)</b>	<p><b>CFO:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>Allows entering a positive or a negative client frequency offset in ppm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:COFF:FREQ:OFFS 55</p> <p>SOUR:DATA:TEL:ETH:COFF:FREQ:OFFS?</p> <p>Returns: 55</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOUnt

**:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:OFFSet?**

<b>Description</b>	<p>This command returns a positive or a negative client frequency offset in ppm.</p> <p>At *RST condition, this value is set to 0.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Tx Frequency &gt; Offset value</p> <p>Navigation Path for 1GbE client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Tx Frequency &gt; Offset</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:OFFSet?
<b>Response Syntax</b>	<CFO>
<b>Response(s)</b>	<p><b>CFO:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the entered positive or a negative client frequency offset in ppm.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:COFF:FREQ:OFFS 55</p> <p>SOUR:DATA:TEL:ETH:COFF:FREQ:OFFS?</p> <p>Returns: 55</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOunt?

---

## SCPI Command Reference

### Client Offset

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:ENABle**

<b>Description</b>	<p>This command allows to enable the frequency offset measurements.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Tx Frequency &gt; Offset(checkbox)</p> <p>Navigation Path for 1GbE client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Tx Frequency &gt; Offset(checkbox)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:ENABle <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status of Frequency Offset Analysis.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:COFF:FREQ:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:COFF:FREQ:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABle

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:ENABle?**

<b>Description</b>	<p>This command returns the status of the frequency offset measurements.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Tx Frequency &gt; Offset(checkbox)</p> <p>Navigation Path for 1GbE client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Tx Frequency &gt; Offset(checkbox)</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:ENABle?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of the frequency offset measurements.</p> <p>1 - Enabled</p> <p>0 - Disabled</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:COFF:FREQ:ENAB ON</p> <p>SOUR:DATA:TEL:ETH:COFF:FREQ:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:OTN:FSTRucture:ENABle?

---

## SCPI Command Reference

### Client Offset

---

---

#### :SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:EFRequency?

---

<b>Description</b>	<p>This query returns the frequency of the input signal in bps.</p> <p>At *RST condition, this value is 1.249453922.</p> <p>Navigation Path for Pattern (oduflex mapping) client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Rx Frequency &gt; Expected Frequency</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:EFRequency?
<b>Response Syntax</b>	<Efrequency>
<b>Response(s)</b>	<p><b>Efrequency:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the expected frequency of the input signal.</p>
<b>Example(s)</b>	SENS:DATA:TEL:ETH:COFF:CONF:EFR?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency?

---

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:FOANalysis:ENABle**

---

<b>Description</b>	<p>This command sets the enable and disable status of Frequency Offset Analysis.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Rx Frequency &gt; Frequency Offset Analysis(checkbox)</p>
<b>Syntax</b>	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:FOANalysis:ENABle &lt;wsp&gt;ON   OFF</pre>
<b>Parameter(s)</b>	<p><b>Status:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the status of Frequency Offset Analysis.</p>
<b>Example(s)</b>	<pre>SENS:DATA:TEL:ETH:COFF:CONF:FOAN:ENAB ON SENS:DATA:TEL:ETH:COFF:CONF:FOAN:ENAB? Returns: 1</pre>
<b>See Also</b>	<pre>SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQUency:ENABle</pre>

---

## SCPI Command Reference

### Client Offset

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:FOANalysis:ENABle?**

<b>Description</b>	<p>This query returns the status of Frequency Offset Analysis.</p> <p>At *RST condition, this value is set to ON.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Rx Frequency &gt; Frequency Offset Analysis(checkbox)</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:FOANalysis:ENABle?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the status of Frequency Offset Analysis.</p> <p>1 - Status is enabled.</p> <p>0 - Status is disabled.</p>
<b>Example(s)</b>	<p>SENS:DATA:TEL:ETH:COFF:CONF:FOAN:ENAB ON</p> <p>SENS:DATA:TEL:ETH:COFF:CONF:FOAN:ENAB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency:ENABle?

---



**:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency?**

<b>Description</b>	<p>This query returns the frequency of the input signal.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path for Pattern(oduflex mapping) client: Test(OTN Bert) &gt; Functions &gt; Client Offset &gt; Rx Frequency &gt; Frequency/Offset/MNOFFset/MPOFFset</p>
<b>Syntax</b>	:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency? <wsp>FREQuency   FOFFset   MNOFFset   MPOFFset
<b>Parameter(s)</b>	<p><b>Frequency Type:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: FREQuency   FOFFset   MNOFFset   MPOFFset</p> <p>Returns the frequency of the input signal in bps.</p> <p>FREQuency, Frequency Offset indicates the offset between the expected rate specification and the rate of the input signal.</p> <p>FOFFset, Offset Unit allows the selection of the frequency offset unit. Choices are bps and ppm. The default setting is pm.</p> <p>MNOFFset, Max. Negative Offset indicates the offset between the expected rate specification and the smallest rate recorded from the received signal.</p> <p>MPOFFset, Max. Positive Offset indicates the offset between the expected rate specification and the largest rate recorded from the received signal.</p>
<b>Response Syntax</b>	<Frequency>

## SCPI Command Reference

### Client Offset

---

**:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency?**

#### Response(s)

#### Frequency:

The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element.

Returns the frequency of the input signal in bps.

FREQuency, Frequency Offset indicates the offset between the expected rate specification and the rate of the input signal.

FOFFset, Offset Unit allows the selection of the frequency offset unit. Choices are bps and ppm. The default setting is pm.

MNOFFset, Max. Negative Offset indicates the offset between the expected rate specification and the smallest rate recorded from the received signal.

MPOFFset, Max. Positive Offset indicates the offset between the expected rate specification and the largest rate recorded from the received signal.

#### Example(s)

SENS:DATA:TEL:ETH:COFF:FREQ? FOFF

#### See Also

SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFig:EFRequency?

---

## Traffic Scan

---

:SOURce[1..n]:DATA:TELEcom:TSCan:SCAN:ENABLE

---

<b>Description</b>	<p>This command allows start the scanning according to the configuration.</p> <p>At *RST condition, this value is set to Disabled.</p> <p>Navigation Path: Traffic Gen &amp; Monitoring &gt; Functions&gt; Traffic Scan &gt; Scan</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TSCan:SCAN:ENABLE <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Enable:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Enable/disable the VLAN Scan.</p>
<b>Example(s)</b>	SOUR:DATA:TEL:TSC:SCAN:ENAB 1
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TSCan:SCAN:ENABLE?

---

## SCPI Command Reference

### Traffic Scan

---

**:SOURce[1..n]:DATA:TELEcom:TSCan:SCAN:ENABLE?**

<b>Description</b>	This query returns the scanning status. At *RST condition, this value is set to Disabled. Navigation Path: Traffic Gen & Monitoring > Functions> Traffic Scan > Scan
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TSCan:SCAN:ENABLE?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<b>Set:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Traffic Scan Status.
<b>Example(s)</b>	SOUR:DATA:TEL:TSC:SCAN:ENABLE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TSCan:SCAN:ENABLE

---

---

**:SOURce[1..n]:DATA:TELEcom:TSCan:LEVel:TYPE**

---

<b>Description</b>	<p>This command allows start the scanning according to the configuration.</p> <p>At *RST condition, this value is set to ALL.</p> <p>Navigation Path: Traffic Gen &amp; Monitoring &gt; Functions&gt; Traffic Scan &gt; Level</p>
<b>Syntax</b>	<code>:SOURce[1..n]:DATA:TELEcom:TSCan:LEVel:TYPE &lt;wsp&gt;EVLAN   SVLAN   CVLAN   UNT   ALL</code>
<b>Parameter(s)</b>	<p><b>Level:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: EVLAN   SVLAN   CVLAN   UNT   ALL</p> <p>Sets the level type.</p>
<b>Example(s)</b>	<code>SOUR:DATA:TEL:TSC:LEV:TYPE EVLAN</code>
<b>See Also</b>	<code>SOURce[1..n]:DATA:TELEcom:PATtern:TYPE</code> <code>SOURce[1..n]:DATA:TELEcom:TSCan:LEVel:TYPE?</code>

---

## SCPI Command Reference

### Traffic Scan

---

**:SOURce[1..n]:DATA:TELEcom:TSCan:LEVel:TYPE?**

<b>Description</b>	This query returns the selected Level. At *RST condition, this value is set to ALL. Navigation Path: Traffic Gen & Monitoring > Functions> Traffic Scan > Level
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:TSCan:LEVel:TYPE?
<b>Response Syntax</b>	<Level>
<b>Response(s)</b>	<b>Level:</b> The response data syntax for this parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the level of traffic scan.
<b>Example(s)</b>	SOUR:DATA:TEL:TSC:LEV:TYPE?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATTern:TYPE? SOURce[1..n]:DATA:TELEcom:TSCan:LEVel:TYPE

---

---

**:FETCh[1..n]:DATA:TELEcom:TSCan:LINK:RATE?**

<b>Description</b>	<p>This query returns the link rate.</p> <p>Navigation Path: Traffic Gen &amp; Monitoring &gt; Functions&gt; Traffic Scan &gt; Link Rate</p>
<b>Syntax</b>	<code>:FETCh[1..n]:DATA:TELEcom:TSCan:LINK:RATE? &lt;wsp&gt;EBANDWIDTH   LUTILIZATION</code>
<b>Parameter(s)</b>	<p><b>Rate Layer:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: EBANDWIDTH   LUTILIZATION</p> <p>Link Rate the VLAN Scan.</p> <p>EBANDWIDTH: Ethernet Bandwidth.</p> <p>LUTILIZATION: Line Utilization.</p>
<b>Response Syntax</b>	<code>&lt;Rate&gt;</code>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the VLAN link rate of traffic scan.</p>
<b>Example(s)</b>	<code>FETC:DATA:TEL:TSC:LINK:RATE? LUTILIZATION</code>
<b>See Also</b>	<code>FETCh[1..n]:DATA:TELEcom:TSCan:STATistics:RATE:TOTal?</code>

---

## SCPI Command Reference

### Traffic Scan

---

---

#### :FETCh[1..n]:DATA:TELEcom:TSCan:DISCovered?

---

<b>Description</b>	This query returns the current number of monitored traffic flows discovered value. Navigation Path: Traffic Gen & Monitoring > Functions> Traffic Scan > Discovered
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:TSCan:DISCovered?
<b>Response Syntax</b>	<Discovered>
<b>Response(s)</b>	<b>Discovered:</b> The response data syntax for this parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of discovered value.
<b>Example(s)</b>	FETC:DATA:TEL:TSC:DISC?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:TSCan:LINK:RATE?

---



---

**:FETCh[1..n]:DATA:TELEcom:TSCan:LREached:STATus?**

---

<b>Description</b>	<p>This query returns the status of when resource limit is reached, new traffic flows are no longer created in the discovery results.</p> <p>Navigation Path: Traffic Gen &amp; Monitoring &gt; Functions&gt; Traffic Scan &gt; Limit Reached</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:TSCan:LREached:STATus?
<b>Response Syntax</b>	<Lreached>
<b>Response(s)</b>	<p><b>Lreached:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the VLAN Status of Limit Reached.</p>
<b>Example(s)</b>	FETC:DATA:TEL:TSC:LRE:STAT?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:FIBer:LINK?

---

## SCPI Command Reference

### Traffic Scan

---

**:FETCh[1..n]:DATA:TELEcom:TSCan:LIST?**

<b>Description</b>	<p>This query returns all the output values as per the selection.</p> <p>Navigation Path: Traffic Gen &amp; Monitoring &gt; Functions&gt; Traffic Scan &gt; E-VLAN , S-VALN, C-VLAN and Statistics Table</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:TSCan:LIST? &lt;wsp&gt;EBANDWIDTH   LUTILIZATION[, &lt;Stream number&gt;]</p>
<b>Parameter(s)</b>	<p><b>Rate Layer:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: EBANDWIDTH   LUTILIZATION</p> <p>Starts The Traffic Scan.</p> <p>Received List of scan as per below selection for rate.</p> <p>EBANDWIDTH: Ethernet Bandwidth.</p> <p>LUTILIZATION: Line Utilization.</p> <p><b>Stream number:</b></p> <p>The program data syntax for the second parameter is defined as a &lt;DECIMAL NUMERIC PROGRAM DATA&gt; element.</p> <p>If Stream number is not provided, then the entire VLAN list is returned and if it is provided then particular stream row is returned.</p>
<b>Response Syntax</b>	<p>&lt;List&gt;</p>
<b>Response(s)</b>	<p><b>List:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the Complete List of traffic scan.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:TSC:LIST? LUTILIZATION, 1</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:LOGGer:LIST?</p>

---

**:FETCh[1..n]:DATA:TELEcom:TSCan:STATistics:FCOunt:TOTal?**

---

<b>Description</b>	This query returns total frame count. Navigation Path: Traffic Gen & Monitoring > Functions> Traffic Scan > Total Frame Count
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:TSCan:STATistics:FCOunt:TOTal?
<b>Response Syntax</b>	<Count>
<b>Response(s)</b>	<b>Count:</b> The response data syntax for this parameter is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the total Frame count of traffic scan.
<b>Example(s)</b>	FETC:DATA:TEL:TSC:STAT:FCO:TOT?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:TSCan:STATistics:RATE:TOTal?

---

## SCPI Command Reference

### Traffic Scan

---

**:FETCh[1..n]:DATA:TELEcom:TSCan:STATistics:RATE:TOTal?**

<b>Description</b>	<p>This query returns total rate count.</p> <p>Navigation Path: Traffic Gen &amp; Monitoring &gt; Functions&gt; Traffic Scan &gt; Total Rate</p>
<b>Syntax</b>	<p>:FETCh[1..n]:DATA:TELEcom:TSCan:STATistics:RATE:TOTal? &lt;wsp&gt;EBANDWIDTH   LUTILIZATION</p>
<b>Parameter(s)</b>	<p><b>Rate Layer:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: EBANDWIDTH   LUTILIZATION</p> <p>Total Receive Rate.</p> <p>EBANDWIDTH: Ethernet Bandwidth.</p> <p>LUTILIZATION: Line Utilization.</p>
<b>Response Syntax</b>	<p>&lt;Rate&gt;</p>
<b>Response(s)</b>	<p><b>Rate:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR2 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the total Frame rate of traffic scan.</p>
<b>Example(s)</b>	<p>FETC:DATA:TEL:TSC:STAT:RATE:TOTal? LUTILIZATION</p>
<b>See Also</b>	<p>FETCh[1..n]:DATA:TELEcom:TSCan:LINK:RATE?</p>

## Discover Remote Button

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:CONNect**

---

<b>Description</b>	<p>This command sets the DTS Connection status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;DTS Connect</p> <p>Navigation Path: Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;DTS Connect</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:CONNect &lt;wsp&gt; &lt;Address&gt;</p>
<b>Parameter(s)</b>	<p><b>Address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Set the IP address of the destination.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:REM:CONN 10.192.5.182</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SSUBnet?</p>

---

## SCPI Command Reference

### *Discover Remote Button*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:CONNect?**

<b>Description</b>	<p>This query returns the DTS Connection status.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;DTS Connect</p> <p>Navigation Path: Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;DTS Connect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:CONNect?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the DTS Connection status.</p> <p>CONNECTED, DTS Connection status is enabled.</p> <p>DISCONNECTED, DTS Connection status is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:REM:CONN 10.192.5.182</p> <p>SOUR:DATA:TEL:ETH:REM:CONN?</p> <p>Returns: CONNECTED</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SSUBnet

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:DISConnect**

---

<b>Description</b>	<p>This command disconnects the Dual Test Set.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;DTS Disconnect</p> <p>Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;DTS Disconnect</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:DISConnect
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:REM:CONN 10.192.5.182 SOUR:DATA:TEL:ETH:REM:DISC
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:PATTern:ERRor:PATTern:AUTomated:TYPE?

---

## SCPI Command Reference

### *Discover Remote Button*

---

**:SOURce[1..n]:DATA:TELeom:ETHernet:REMOte:SSUBnet**

<b>Description</b>	<p>This command sets the Scan target for Dual Test Set.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Scan</p> <p>Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Scan</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELeom:ETHernet:REMOte:SSUBnet <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the Scan target statue for Dual Test Setup</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:REM:SSUB ON
<b>See Also</b>	SOURce[1..n]:DATA:TELeom:ETHernet:REMOte:SCANtarget:TYPE?

---



**:SOURce[1..n]:DATA:TELEcom:ETHernet:REMOte:SSUBnet?**

<b>Description</b>	<p>This query returns the Scan target for Dual Test Set.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Scan</p> <p>Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Scan</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:REMOte:SSUBnet?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the transmitter status.</p> <p>1, Scanning status is enabled.</p> <p>0, Scanning status is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:REM:SSUB ON</p> <p>SOUR:DATA:TEL:ETH:REM:SSUB?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:REMOte:SCANtarget:TYPE

## SCPI Command Reference

### *Discover Remote Button*

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SCANtarget:TYPE**

<b>Description</b>	<p>This command sets the Scan target type.</p> <p>At *RST condition, this value is set to SUBNET.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Target</p> <p>Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Target</p>
<b>Syntax</b>	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SCANtarget:TYPE &lt;wsp&gt;SUBNET   SPECIFICIP</p>
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;CHARACTER PROGRAM DATA&gt; element.</p> <p>The allowed elements for this parameter are: SUBNET   SPECIFICIP</p> <p>Sets the Scan target type.</p> <p>SUBNET: SUBNET</p> <p>SPECIFICIP: SPECIFICIP</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:REM:SCAN:TYPE SUBNET</p>
<b>See Also</b>	<p>SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SSUBnet?</p>

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SCANtarget:TYPE?**

<b>Description</b>	<p>This query returns the Scan target.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Target</p> <p>Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Target</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SCANtarget:TYPE?
<b>Response Syntax</b>	<Set>
<b>Response(s)</b>	<p><b>Set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the scan target type</p> <p>SUBNET, returns the SUBNET as type of scan target</p> <p>SPECIFICIP, returns the SPECIFICIP as type of scan target</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:REM:SCAN:TYPE SUBNET</p> <p>SOUR:DATA:TEL:ETH:REM:SCAN:TYPE?</p> <p>Returns: SUBNET</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:SSUBnet

## SCPI Command Reference

### *Discover Remote Button*

---

**:SOURce[1..n]:DATA:TELecom:ETHernet:REMOte:LOOP:UP**

<b>Description</b>	<p>This command sets the Loop up to smart loopback test application.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Loop Up</p> <p>Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Loop Up</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELecom:ETHernet:REMOte:LOOP:UP <wsp> <Address>
<b>Parameter(s)</b>	<p><b>Address:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;STRING PROGRAM DATA&gt; element.</p> <p>Set the IP address of the loopup</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:REM:LOOP:UP 10.192.5.182
<b>See Also</b>	SOURce[1..n]:DATA:TELecom:ETHernet:REMOte:CONNect?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:LOOP:DOWN**

---

<b>Description</b>	<p>This command sets the Loop down for smart loopback test application.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Loop Down</p> <p>Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery&gt;Loop Down</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:LOOP:DOWN
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:REM:LOOP:UP 10.192.5.182 SOUR:DATA:TEL:ETH:REM:LOOP:DOWN
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:REMote:DISConnect

---

## SCPI Command Reference

### *Discover Remote Button*

---

---

#### :SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:CStatus

---

<b>Description</b>	<p>This command enables/disables the loopback mode type for Smart Loopback application. At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test Setup &gt; Test Configurator &gt; RFC 2544 &gt;Global Options&gt;Coupled</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:CStatus <wsp>ON   OFF
<b>Parameter(s)</b>	<p><b>Set:</b></p> <p>The program data syntax for the first parameter is defined as a &lt;Boolean Program Data&gt; element.</p> <p>The allowed elements for this parameter are: ON   OFF</p> <p>Sets the coupled status</p>
<b>Example(s)</b>	SOUR:DATA:TEL:ETH:RFC:CST ON
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:NETWork:VLAN:STATus?

---

---

**:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:CStatus?**

---

<b>Description</b>	<p>This query enables/disables the loopback mode type for Smart Loopback application.</p> <p>At *RST condition, this value is set to OFF.</p> <p>Navigation Path: Test &gt; Setup &gt; Test Configurator &gt; RFC 2544 &gt;Global Options&gt;Coupled</p>
<b>Syntax</b>	:SOURce[1..n]:DATA:TELEcom:ETHernet:RFC:CStatus?
<b>Response Syntax</b>	<set>
<b>Response(s)</b>	<p><b>set:</b></p> <p>The response data syntax for this parameter is defined as a &lt;NR1 NUMERIC RESPONSE DATA&gt; element.</p> <p>Returns the coupled status</p> <p>1, coupled status is enabled.</p> <p>0, coupled status is disabled.</p>
<b>Example(s)</b>	<p>SOUR:DATA:TEL:ETH:RFC:CST ON</p> <p>SOUR:DATA:TEL:ETH:RFC:CST?</p> <p>Returns: 1</p>
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:ETHernet:SLOopback:NETWork:VLAN:STATus

---

## SCPI Command Reference

### *Discover Remote Button*

---

---

#### **:FETCh[1..n]:DATA:TELEcom:ETHernet:DUALtest:STATistics?**

---

<b>Description</b>	<p>This query returns the IP address, Remote ID, Capability and Status of remote connection.</p> <p>This command is an event and is not associated with an *RST condition or a query form.</p> <p>Navigation Path: Test Setup &gt; RFC 2544&gt; Test Configurator &gt; RFC 2544&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery</p> <p>Test Setup &gt;EtherSAM (Y.1564) &gt; Test Configurator &gt; EtherSAM&gt; Global&gt; Dual Test Set&gt;Discover Remote&gt;Remote Modules Discovery</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:DUALtest:STATistics?
<b>Response Syntax</b>	<Value>
<b>Response(s)</b>	<p><b>Value:</b></p> <p>The response data syntax for this parameter is defined as a &lt;STRING RESPONSE DATA&gt; element.</p> <p>Returns the IP address, Remote ID, Capability and Status of remote connection.</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:DUAL:STAT?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:RFC:LATency:FCOut:RX?

---



**:FETCh[1..n]:DATA:TELEcom:ETHernet:REMote:RSCStatus?**

<b>Description</b>	<p>This query returns DTS Connection is established, an Icon shall appear within the Status Bar on both local and remote modules. The icon shall be personalized in function of the selected Remote Capability.</p> <p>At *RST condition, this value is device dependent.</p> <p>Navigation Path: Test &gt;Status Bar(EtherSAM/EtherBert/SMARTLoopBack/RFC/TGEN)</p>
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:ETHernet:REMote:RSCStatus?
<b>Response Syntax</b>	<Status>
<b>Response(s)</b>	<p><b>Status:</b></p> <p>The response data syntax for this parameter is defined as a &lt;CHARACTER RESPONSE DATA&gt; element.</p> <p>Returns the DTS connection status personalized in function of the selected Remote Capability</p> <p>IDLE,status is idle</p> <p>CONNECTEDTOREMOTE,Connected to remote</p> <p>CONTROLLEDBYREMOTE,controlled by remote</p>
<b>Example(s)</b>	FETC:DATA:TEL:ETH:REM:RSCS?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:ETHernet:PORT:GLOBal:IPV:STATus?

## Lpbk Tool Button (Interface)

---

:FETCh[1..n]:DATA:TELEcom:MODule:DETailS:MID?

---

<b>Description</b>	This query returns the Module Id. At *RST condition, this value is set to device-dependent. Navigation Path: About > Module Details > Module Id
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:MODule:DETailS:MID?
<b>Response Syntax</b>	<ModuleId>
<b>Response(s)</b>	<b>ModuleId:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the Module Id.
<b>Example(s)</b>	FETC:DATA:TEL:MOD:DET:MID?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:TEST:STATus?

---

---

**:FETCh[1..n]:DATA:TELEcom:MODUle:DETAils:SNUMber?**

---

<b>Description</b>	This query returns the Module Serial Number. At *RST condition, this value is set to device-dependent. Navigation Path: About > Module Details > Serial Number
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:MODUle:DETAils:SNUMber?
<b>Response Syntax</b>	<Serial Number>
<b>Response(s)</b>	<b>Serial Number:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the Module serial number.
<b>Example(s)</b>	FETC:DATA:TEL:MOD:DET:SNUM?
<b>See Also</b>	SOURce[1..n]:DATA:TELEcom:TIMer:UDEF? SOURce[1..n]:DATA:TELEcom:TIMer:DURATION

---

## SCPI Command Reference

### *Lpbk Tool Button (Interface)*

---

---

#### :FETCh[1..n]:DATA:TELEcom:MODUle:DETAils:AHRevision?

---

<b>Description</b>	This query returns the Assembly Hardware Revision. At *RST condition, this value is set to device-dependent. Navigation Path: About > Module Details > Assembly Hardware Revision
<b>Syntax</b>	:FETCh[1..n]:DATA:TELEcom:MODUle:DETAils:AHRevision?
<b>Response Syntax</b>	<Revision>
<b>Response(s)</b>	<b>Revision:</b> The response data syntax for this parameter is defined as a <STRING RESPONSE DATA> element. Returns the Assembly hardware revision.
<b>Example(s)</b>	FETC:DATA:TEL:MOD:DET:AHR?
<b>See Also</b>	FETCh[1..n]:DATA:TELEcom:TEST:STATus?

---

# 10 *Obsolete SCPI Commands*

This section lists obsolete commands and their replacement commands when applicable.

Obsolete Command	Replacement Command
:SOURce[1..n]:DATA:TELecom:TAPPLication:TEST:TYPE	:SOURce[1..n]:DATA:TELecom:TEST:TYPE
:SOURce[1..n]:DATA:TELecom:TAPPLication:TEST:TYPE?	:SOURce[1..n]:DATA:TELecom:TEST:TYPE?
:SOURce[1..n]:DATA:TELecom:SONet:TEST:TYPE	
:SOURce[1..n]:DATA:TELecom:SONet:TEST:TYPE?	

[www.EXFO.com](http://www.EXFO.com) · [info@exfo.com](mailto:info@exfo.com)

<b>CORPORATE HEADQUARTERS</b>	400 Godin Avenue	Quebec (Quebec) G1M 2K2 CANADA Tel.: 1 418 683-0211 · Fax: 1 418 683-2170
<b>EXFO AMERICA</b>	3400 Waterview Parkway Suite 100	Richardson, TX 75080 USA Tel.: 1 972-761-9271 · Fax: 1 972-761-9067
<b>EXFO EUROPE</b>	Winchester House, School Lane	Chandlers Ford, Hampshire S053 4DG ENGLAND Tel.: +44 2380 246 800 · Fax: +44 2380 246 801
<b>EXFO ASIA-PACIFIC</b>	62 Ubi Road 1, #09-01/02 Oxley Bizhub 2	SINGAPORE 408734 Tel.: +65 6333 8241 · Fax: +65 6333 8242
<b>EXFO CHINA</b>	Beijing Global Trade Center, Tower C, Room 1207, 36 North Third Ring Road East, Dongcheng District	Beijing 100013 P. R. CHINA Tel.: +86 (10) 5825 7755 · Fax: +86 (10) 5825 7722
<b>EXFO SERVICE ASSURANCE</b>	270 Billerica Road	Chelmsford MA, 01824 USA Tel.: 1 978 367-5600 · Fax: 1 978 367-5700
<b>EXFO FINLAND</b>	Elektroniikkatie 2	FI-90590 Oulu, FINLAND Tel.: +358 (0) 403 010 300 · Fax: +358 (0) 8 564 5203
<b>TOLL-FREE</b>	(USA and Canada)	1 800 663-3936

© 2017 EXFO Inc. All rights reserved.  
Printed in Canada (2017-02)

