

SONET/SDH SCPI COMMANDS

FTB/IQS-81xx Transport Blazer for FTB-400, FTB-500 and
IQS-600 Platforms

NETWORK TESTING

REFERENCE GUIDE



Copyright © 2007–2015 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.

Contents

1	Introducing the FTB/IQS-81xx Transport Blazer SCPI Commands	1
	Key Features of FTB/IQS-8105 Transport Blazer	3
	Key Features of FTB/IQS-8115 Transport Blazer	3
	Key Features of FTB/IQS-8120/8130 Transport Blazer	4
	Key Features of FTB/IQS-8120NGE/8130NGE Transport Blazer	4
	Key Features of FTB/IQS-8140 Transport Blazer	5
	Conventions	7
2	Getting Started with FTB/IQS-81xx Transport Blazer SCPI Commands	9
	Selecting the Network Analyzer Mode	9
	Using SCPI Commands to create SONET OC-48 Test	11
3	Introducing the SCPI over TCP/IP Service	15
	Features	15
	Steps to execute SCPI commands through SCPI over TCP/IP Service	17
	Scenarios for release/block of module(s)	20
	Scenario 1: Block	20
	Scenario 2: Release	20
	Internal (Protocol) Commands of SCPI over TCP/IP Service	25
	Execution of single command	25
	Execution of multiple commands (BEGIN and END block)	25
	ABORT BEGIN	26
	CLOSE	27
	STATUS CLIENT	27
	STATUS MODULE	28
	STATUS CONNECTION	28

Contents

CONNECT LINS	29
CLOSE LINS.....	30
KILL LINS	31
WHO M I?	32
CLEAR LOGS	32
3 SONET/SDH SCPI Command Reference	33
Network Analyzer	34
:INSTrument[1..n]:SElect	34
:INSTrument[1..n]:SElect?	36
SONET/SDH Analyzer	38

Test Setup Command Reference	38
:SOURce[1..n]:DATA:TELEcom:MODE	38
:SOURce[1..n]:DATA:TELEcom:MODE?	39
:OUTPut[1..n]:TELEcom:CONNector	40
:OUTPut[1..n]:TELEcom:CONNector?	42
:SOURce[1..n]:DATA:TELEcom:INTerface:TYPE	44
:SOURce[1..n]:DATA:TELEcom:INTerface:TYPE?	48
:SOURce[1..n]:DATA:TELEcom:HOP:TYPE	52
:SOURce[1..n]:DATA:TELEcom:HOP:TYPE?	55
:SOURce[1..n]:DATA:TELEcom:LOP:TYPE	57
:SOURce[1..n]:DATA:TELEcom:LOP:TYPE?	59
:SOURce[1..n]:DATA:TELEcom:ODU:TYPE	61
:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?	68
:SOURce[1..n]:DATA:TELEcom:MAPPing:TYPE	74
:SOURce[1..n]:DATA:TELEcom:MAPPing:TYPE?	76
:SOURce[1..n]:DATA:TELEcom:POSition	78
:SOURce[1..n]:DATA:TELEcom:POSition?	81
:SOURce[1..n]:CLOCK:SOURce	83
:SOURce[1..n]:CLOCK:SOURce?	84
:SOURce[1..n]:DATA:TELEcom:ELECtrical:CODE	85
:SOURce[1..n]:DATA:TELEcom:ELECtrical: CODE?	87
:SENSe[1..n]:DATA:TELEcom:ELECtrical:CODE	89
:SENSe[1..n]:DATA:TELEcom:ELECtrical:CODE?	91
:SOURce[1..n]:DATA:TELEcom:ELECtrical:LBO	93
:SOURce[1..n]:DATA:TELEcom:ELECtrical:LBO?	95
:SENSe[1..n]:DATA:TELEcom:ELECtrical: TERMination	97
:SENSe[1..n]:DATA:TELEcom:ELECtrical: TERMination?	99
:OUTPut[1..n]:TELEcom:CODE	101
:OUTPut[1..n]:TELEcom:CODE?	102
:OUTPut[1..n]:TELEcom:FRAMing	103
:OUTPut[1..n]:TELEcom:FRAMing?	105
:OUTPut[1..n]:TELEcom:LEVel	107
:OUTPut[1..n]:TELEcom:LEVel?	109
:OUTPut[1..n]:TELEcom:TERMination	111
:OUTPut[1..n]:TELEcom:TERMination?	112
:SOURce[1..n]:DATA:TELEcom:TEST	113
:SOURce[1..n]:DATA:TELEcom:TEST?	114
:SOURce[1..n]:DATA:TELEcom:RESet	115
:SOURce[1..n]:DATA:TELEcom:HRESet	116
:SOURce[1..n]:DATA:TELEcom:MOUNt	117
:SOURce[1..n]:DATA:TELEcom:CLear	118
:CONFig[1..n]:DATA:TELEcom:LOAD	119

Contents

:CONFig[1..n]:DATA:TELEcom:SAVE	120
:SOURce[1..n]:DATA:TELEcom:BACKground: OTN	121
:SOURce[1..n]:DATA:TELEcom:BACKground: OTN?	123
:SOURce[1..n]:DATA:TELEcom:BACKground: OTNFlex	125
:SOURce[1..n]:DATA:TELEcom:BACKground: OTNFlex?	127
:SOURce[1..n]:DATA:TELEcom:BACKground: SDHSonet:HOP	129
:SOURce[1..n]:DATA:TELEcom:BACKground: SDHSonet:HOP?	131
:SOURce[1..n]:DATA:TELEcom:BACKground: SDHSonet:LOP	133
:SOURce[1..n]:DATA:TELEcom:BACKground: SDHSonet:LOP?	135
:SOURce[1..n]:DATA:TELEcom:BACKground: PDH	137
:SOURce[1..n]:DATA:TELEcom:BACKground: PDH?	138
:SOURce[1..n]:DATA:TELEcom:BACKground: BULK	140
:SOURce[1..n]:DATA:TELEcom:BACKground: BULK?	141
:SOURce[1..n]:DATA:TELEcom:BACKground: COMPUtation	142
:SOURce[1..n]:DATA:TELEcom:BACKground: COMPUtation?	143
:FETCh[1..n]:DATA:TELEcom:TEST:TIME?	145
:FETCh[1..n]:DATA:TELEcom:TEST:GLOBal: HISTory?	146
:FETCh[1..n]:DATA:TELEcom:TEST:GLOBal: CURRent?	148
:SENSe[1..n]:DATA:TELEcom:TEST:STARt:TIME	150
:SENSe[1..n]:DATA:TELEcom:TEST:STARt:TIME?	151
:SENSe[1..n]:DATA:TELEcom:TEST:STARt: TIME:ENABled	152
:SENSe[1..n]:DATA:TELEcom:TEST:STARt: TIME:ENABled?	153
:SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME	154
:SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME?	155
:SENSe[1..n]:DATA:TELEcom:TEST:STOP: TIME:ENABled	156
:SENSe[1..n]:DATA:TELEcom:TEST:STOP: TIME:ENABled?	157
:SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME	158
:SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME?	160
:SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME:ENABled	162
:SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME:ENABled?	163
:SENSe[1..n]:DATA:TELEcom:TEST:DURation: USER	164
:SENSe[1..n]:DATA:TELEcom:TEST:DURation: USER?	165
:SENSe[1..n]:DATA:TELEcom:TEST:TIMER:RESet	167
:SENSe[1..n]:DATA:TELEcom:TEST:TIMER: STATus?	168
:SOURce[1..n]:DATA:TELEcom:COUPled	169
:SOURce[1..n]:DATA:TELEcom:COUPled?	170
:SOURce[1..n]:DATA:TELEcom:THRough	171
:SOURce[1..n]:DATA:TELEcom:THRough?	172
:SOURce[1..n]:DATA:TELEcom:INTRusive	173
:SOURce[1..n]:DATA:TELEcom:INTRusive?	174
:SOURce[1..n]:DATA:TELEcom:INTRusive:OTN	175
:SOURce[1..n]:DATA:TELEcom:INTRusive:OTN?	176

:SOURce[1..n]:DATA:TELEcom:TSL	177
:SOURce[1..n]:DATA:TELEcom:TSL?	180
:SOURce[1..n]:DATA:TELEcom:TEST:NODE: DIRection	182
:SOURce[1..n]:DATA:TELEcom:TEST:NODE: DIRection?	184
:SOURce[1..n]:DATA:TELEcom:OTN:TRIButary: PORT	186
:SOURce[1..n]:DATA:TELEcom:OTN:TRIButary: PORT?	189
Clock Synchronization Command Reference	191
:OUTPut[1..n]:TELEcom:CODE	191
:OUTPut[1..n]:TELEcom:CODE?	193
:OUTPut[1..n]:TELEcom:FRAMing	195
:OUTPut[1..n]:TELEcom:FRAMing?	198
:OUTPut[1..n]:TELEcom:LEVel	200
:OUTPut[1..n]:TELEcom:LEVel?	202
:OUTPut[1..n]:TELEcom:TERMination	204
:OUTPut[1..n]:TELEcom:TERMination?	205
:OUTPut[1..n]:TELEcom:CLOCK:ALARm: STATus?	206
:OUTPut[1..n]:TELEcom:CLOCK:FREQuency?	208
:OUTPut[1..n]:TELEcom:CLOCK:FREQuency: OFFSet?	209
:INPut[1..n]:TELEcom:CODE	210
:INPut[1..n]:TELEcom:CODE?	211
:INPut[1..n]:TELEcom:FRAMing	212
:INPut[1..n]:TELEcom:FRAMing?	214
:INPut[1..n]:TELEcom:LEVel	216
:INPut[1..n]:TELEcom:LEVel?	217
:INPut[1..n]:TELEcom:CLOCK	219
:INPut[1..n]:TELEcom:CLOCK?	221
:INPut[1..n]:TELEcom:LBO	223
:INPut[1..n]:TELEcom:LBO?	225
:INPut[1..n]:TELEcom:CLOCK:ALARm: STATus?	227
:INPut[1..n]:TELEcom:CLOCK:SIGNaL: STATus?	229
:INPut[1..n]:TELEcom:BACKplane:CLOCK	231
:INPut[1..n]:TELEcom:BACKplane:CLOCK?	233
:INPut[1..n]:TELEcom:BACKplane	235
:INPut[1..n]:TELEcom:BACKplane?	236
:INPut[1..n]:TELEcom:BACKplane:ALARm: STATus?	237
:OUTPut[1..n]:TELEcom:REFoutput: FREQuency?	239
:OUTPut[1..n]:TELEcom:REFoutput: SIGNaL: STATus?	240
:OUTPut[1..n]:TELEcom:REFoutput: DRATio	242
:OUTPut[1..n]:TELEcom:REFoutput: DRATio?	244

Contents

Port Command Reference	246
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency?	246
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:NOMinal?	247
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:OFFSet:VALue	248
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:OFFSet:VALue?	250
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:OFFSet	252
:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:OFFSet?	253
:SENSe[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency?	254
:SENSe[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:POSitive?	255
:SENSe[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:NEGative?	257
:SENSe[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:OFFSet:VALue?	259
:FETCh[1..n]:DATA:TELEcom:OPTical:INPut: STATus?	261
:FETCh[1..n]:DATA:TELEcom:OPTical:OUTPut: STATus?	262
:SENSe[1..n]:DATA:TELEcom:OPTical:PORT: PLEVel?	263
:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:TYPE	264
:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:TYPE?	265
:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT	266
:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT?	268
:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:HISTory?	269
:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:CURRent?	271
:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:SEConds?	273
:OUTPut[1..n]:TELEcom:LASer	275
:OUTPut[1..n]:TELEcom:LASer?	276
:OUTPut[1..n]:TELEcom:LASer:WAVelength?	277
:FETCh[1..n]:DATA:TELEcom:ELECtrical: OUTPut:STATus?	278
:FETCh[1..n]:DATA:TELEcom:ELECtrical: INPut:STATus?	279
:SOURce[1..n]:DATA:TELEcom:ELECtrical:PORT:FREQUency?	280
:SOURce[1..n]:DATA:TELEcom:ELECtrical:PORT:FREQUency:NOMinal?	281
:SOURce[1..n]:DATA:TELEcom:ELECtrical:PORT:FREQUency:OFFSet:VALue	282
:SOURce[1..n]:DATA:TELEcom:ELECtrical:PORT:FREQUency:OFFSet:VALue?	284
:SOURce[1..n]:DATA:TELEcom:ELECtrical:PORT:FREQUency:OFFSet	286
:SOURce[1..n]:DATA:TELEcom:ELECtrical:PORT:FREQUency:OFFSet?	288
:SENSe[1..n]:DATA:TELEcom:ELECtrical:PORT: PLEVel?	290
:SENSe[1..n]:DATA:TELEcom:ELECtrical:PORT: FREQUency?	291
:SENSe[1..n]:DATA:TELEcom:ELECtrical:PORT: FREQUency:NEGative?	292
:SENSe[1..n]:DATA:TELEcom:ELECtrical:PORT: FREQUency:OFFSet:Value?	294
:SENSe[1..n]:DATA:TELEcom:ELECtrical:PORT: FREQUency:POSitive?	296
:FETCh[1..n]:DATA:TELEcom:ELECtrical:PORT: AMPLitude?	298
:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:TYPE	299
:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:TYPE?	301
:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT	303
:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT?	305

:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:MANual:TYPE	307
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:MANual:TYPE?	309
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AMOUnt	311
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AMOUnt?	313
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:INJect	315
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:TYPE	316
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:TYPE?	318
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:RATE	320
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:RATE?	322
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated	324
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated?	326
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:CONTInuous	328
:SOURCE[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:CONTInuous?	330
:FETCh[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:HISTory?	332
:FETCh[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:CURRent?	334
:FETCh[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:SECOnds?	336
:FETCh[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:HISTory?	338
:FETCh[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:SECOnds?	340
:FETCh[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:CURRent?	342
:FETCh[1..n]:DATA:TELEcom:ELECtrical:ERRor:PORT:COUNT?	344
:FETCh[1..n]:DATA:TELEcom:ELECtrical:ERRor: PORT:RATE?	346

Contents

Pattern Command Reference	348
:SOURCE[1..n]:DATA:TELEcom:PATtern:TYPE	348
:SOURCE[1..n]:DATA:TELEcom:PATtern:TYPE?	351
:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE	353
:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE?	356
:SOURCE[1..n]:DATA:TELEcom:PATtern: COUPled	358
:SOURCE[1..n]:DATA:TELEcom:PATtern: COUPled?	359
:SENSe[1..n]:DATA:TELEcom:PATtern:COUPled	361
:SENSe[1..n]:DATA:TELEcom:PATtern: COUPled?	362
:SOURCE[1..n]:DATA:TELEcom:POLarity	363
:SOURCE[1..n]:DATA:TELEcom:POLarity?	364
:SENSe[1..n]:DATA:TELEcom:POLarity	366
:SENSe[1..n]:DATA:TELEcom:POLarity?	367
:SOURCE[1..n]:DATA:TELEcom:PATtern: OVERwrite:ENABLEd	368
:SOURCE[1..n]:DATA:TELEcom:PATtern: OVERwrite:ENABLEd?	369
:SOURCE[1..n]:DATA:TELEcom:PATtern:TYPE: USER:VALue	370
:SOURCE[1..n]:DATA:TELEcom:PATtern:TYPE: USER:VALue?	372
:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE: USER:VALue	374
:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE: USER:VALue?	376
:SENSe[1..n]:DATA:TELEcom:PATtern:TRAFfic	378
:SENSe[1..n]:DATA:TELEcom:PATtern:TRAFfic?	379
:SOURCE[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern:TYPE	380
:SOURCE[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern:TYPE?	381
:SOURCE[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern	382
:SOURCE[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern?	383
:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:MANual:TYPE	384
:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:MANual:TYPE?	385
:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AMOUnt	386
:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AMOUnt?	388
:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:INJect	390
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:HISTory?	391
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:SECOnds?	393
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:CURRent?	395
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:COUnt?	397
:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:RATE?	399
:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern:HISTory?	401
:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern:SECOnds?	403
:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern:CURRent?	405
:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTOmated:TYPE	407
:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTOmated:TYPE?	409
:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTOmated:RATE	411
:SOURCE[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTOmated:RATE?	413

:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated	415
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated?	417
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:CONTInuous	419
:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:CONTInuous?	421
:FETCh[1..n]:DATA:TELEcom:PATtern:PM: STATistics?	423

SONET Command Reference	426
:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:ENABled	426
:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:ENABled?	427
:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:OVERwrite:ENABled	428
:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:OVERwrite:ENABled?	430
:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern	432
:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern?	434
:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern:B16	436
:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern:B16?	438
:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern:B64	440
:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern:B64?	442
:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:J:TIM	444
:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:J:TIM?	445
:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:J:TIM:PATtern	446
:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:J:TIM:PATtern?	448
:FETCh[1..n]:DATA:TELEcom:SONet:OVERhead:J:TIM:PATtern:RECEived?	450
:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:J:TIM:PATtern:B16	452
:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:J:TIM:PATtern:B16?	453
:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:J:TIM:PATtern:B64	455
:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:J:TIM:PATtern:B64?	457
:SOURce[1..n]:DATA:TELEcom:SONet:HOP: PATH:LABel	459
:SOURce[1..n]:DATA:TELEcom:SONet:HOP: PATH:LABel?	464
:SENSe[1..n]:DATA:TELEcom:SONet:HOP:PATH:LABel:EXPEcted	468
:SENSe[1..n]:DATA:TELEcom:SONet:HOP:PATH:LABel:EXPEcted?	471
:FETCh[1..n]:DATA:TELEcom:SONet:HOP:PATH: LABel?	474
:SENSe[1..n]:DATA:TELEcom:SONet:HOP: PUNeq	475
:SENSe[1..n]:DATA:TELEcom:SONet:HOP: PUNeq?	476
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:TYPE	477
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:TYPE?	479
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:RATE	481
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:RATE?	483
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated	485
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated?	487
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:CONTInuous	489
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:CONTInuous?	491
:SOURce[1..n]:DATA:TELEcom:SONet:LOP: PATH:LABel	493
:SOURce[1..n]:DATA:TELEcom:SONet:LOP: PATH:LABel?	495
:SENSe[1..n]:DATA:TELEcom:SONet:LOP:PATH: LABel:EXPEcted	497
:SENSe[1..n]:DATA:TELEcom:SONet:LOP:PATH: LABel:EXPEcted?	499
:FETCh[1..n]:DATA:TELEcom:SONet:LOP:PATH: LABel?	501
:SENSe[1..n]:DATA:TELEcom:SONet:LOP: PUNeq	503
:SENSe[1..n]:DATA:TELEcom:SONet:LOP: PUNeq?	505

:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:TYPE	507
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:TYPE?	509
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:RATE	511
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:RATE?	513
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated	515
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated?	517
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:CONTInuous	519
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:CONTInuous?	521
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:TYPE	523
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:TYPE?	524
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon	526
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon?	527
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:MANual:TYPE	528
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:MANual:TYPE?	529
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AMOUnt	530
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AMOUnt?	532
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:INJect	534
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:HISTory?	535
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:SECOnds?	537
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:CURREnt?	539
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:HISTory?	541
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:SECOnds?	543
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:CURREnt?	545
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:COUnt?	547
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:RATE?	549
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated:TYPE	551
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated:TYPE?	553
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated:RATE	555
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated:RATE?	557
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated	559
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated?	561
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated:CONTInuous	563
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated:CONTInuous?	565
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:TYPE	567
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:TYPE?	569
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE	570
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE?	571
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LINE:MANual:TYPE	572
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LINE:MANual:TYPE?	573
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AMOUnt	574
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AMOUnt?	576
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LINE:INJect	578

Contents

:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LINE:HISTory?	579
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LINE:SEConds?	581
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LINE:CURRent?	583
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: LINE:HISTory?	585
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: LINE:SEConds?	587
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: LINE:CURRent?	589
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: LINE:COUnT?	591
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: LINE:RATE?	593
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:TYPE	595
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:TYPE?	597
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:RATE	599
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:RATE?	601
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated	603
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated?	605
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:CONTInuous	607
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:CONTInuous?	609
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:BURSt:TYPE	611
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:BURSt:TYPE?	612
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:BURSt:MODE	613
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:BURSt:MODE?	614
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:BURSt:DURation	616
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:BURSt:DURation?	618
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:BURSt:PERiod	620
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:BURSt:PERiod?	622
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:BURSt	624
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:BURSt?	626
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt:TYPE	628
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt:TYPE?	629
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt:MODE	631
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt:MODE?	633
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt:DURation	635
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt:DURation?	637
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt:PERiod	639
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt:PERiod?	641
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt	643
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt?	645
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE	647
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE?	648
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:MODE	650
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:MODE?	652
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:DURation	654
:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:DURation?	656

:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:PERiod	658
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:PERiod?	660
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt	662
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt?	664
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE	666
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE?	667
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:MODE	669
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:MODE?	671
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:DURation	673
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:DURation?	675
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt	677
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt?	678
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:PERiod	680
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:PERiod?	682
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:TYPE	684
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:TYPE?	685
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:MODE	687
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:MODE?	689
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:DURation	691
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:DURation?	693
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:PERiod	695
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:PERiod?	697
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt	699
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt?	701
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:TYPE	703
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:TYPE?	706
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:MODE	708
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:MODE?	710
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:DURation	712
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:DURation?	714
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:PERiod	716
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:PERiod?	718
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt	720
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt?	722
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:TYPE	724
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:TYPE?	726
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:MODE	728
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:MODE?	730
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:DURation	732
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:DURation?	734
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:PERiod	736
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:PERiod?	738

Contents

:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt	740
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt?	742
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:TYPE	744
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:TYPE?	747
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:MODE	750
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:MODE?	752
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:DUration	754
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:DUration?	756
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:PERiod	758
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:PERiod?	760
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt	762
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt?	764
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:TYPE	766
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:TYPE?	769
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH	771
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH?	772
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:MANual:TYPE	774
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:MANual:TYPE?	776
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AMOUNT	778
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AMOUNT?	780
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:INJect	782
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:HISTory?	783
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:SEConds?	785
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:CURRent?	787
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:COUNt?	789
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:RATE?	791
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:HISTory?	793
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:SEConds?	797
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:CURRent?	801
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:TYPE	805
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:TYPE?	807
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH	809
:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH?	811
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:MANual:TYPE	813
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:MANual:TYPE?	815
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AMOUNT	817
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AMOUNT?	819
:SOURCE[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:INJect	821
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:HISTory?	822
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:SEConds?	824
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP: PATH:CURRent?	826
:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP: PATH:COUNt?	828

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP: PATH:RATE?	830
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:HISTory?	832
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:SEConds?	836
:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:CURRent?	839
:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTion:OVERhead	843
:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTion:OVERhead?	846
:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTion:DISabled	849
:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTion:OVERwrite:ENABLEd	850
:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTion:OVERwrite:ENABLEd?	853
:FETCh[1..n]:DATA:TELEcom:SONet:OH: SECTion:OVERwrite:STATus?	856
:SENSe[1..n]:DATA:TELEcom:SONet:OH: SECTion:OVERhead?	857
:SOURce[1..n]:DATA:TELEcom:SONet:OH: LINE:OVERhead	860
:SOURce[1..n]:DATA:TELEcom:SONet:OH: LINE:OVERhead?	864
:SOURce[1..n]:DATA:TELEcom:SONet:OH: LINE:DISabled	867
:SOURce[1..n]:DATA:TELEcom:SONet:OH: LINE:OVERwrite:ENABLEd	868
:SOURce[1..n]:DATA:TELEcom:SONet:OH: LINE:OVERwrite:ENABLEd?	872
:FETCh[1..n]:DATA:TELEcom:SONet:OH: LINE: OVERwrite:STATus?	876
:SENSe[1..n]:DATA:TELEcom:SONet:OH: LINE: OVERhead?	877
:FETCh[1..n]:DATA:TELEcom:SONet:SECTion: PM:STATistics?	880
:FETCh[1..n]:DATA:TELEcom:SONet:LINE:PM: STATistics?	883

SDH/SONET Common Command Reference	886
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:ENABled	886
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:ENABled?	887
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:OVERwrite:ENABled	888
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:OVERwrite:ENABled?	890
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern	892
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern?	894
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern:B16	896
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern:B16?	898
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern:B64	900
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern:B64?	902
:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM	904
:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM?	905
:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern	906
:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern?	908
:FETCh[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:RECEived?	910
:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B16	912
:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B16?	914
:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B64	916
:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B64?	918
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:VALue?	920
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:SIZE	921
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:SIZE?	923
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement:SIZE	925
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement:SIZE?	927
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement	929
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement	930
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:VALue	931
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:VALue?	933
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW	935
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:FLAG	936
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:FLAG?	937
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:VALue?	938
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:OFFSet?	939
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:COUNT?	940
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement:COUNT?	941
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:SEConds?	943
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement:SEConds?	944
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NDF:COUNT?	945
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NNDF:COUNT?	946
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NDF:SEConds?	947
:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NNDF:SEConds?	948

:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:VALue?	949
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:SIZE	950
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:SIZE?	952
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement:SIZE	954
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement:SIZE?	956
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement	958
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement	959
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW:VALue	960
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW:VALue?	962
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW	964
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW:FLAG	965
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW:FLAG?	967
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:VALue?	969
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:OFFSet?	970
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:COUNT?	971
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement:COUNT?	973
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:SEConds?	975
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement:SEConds?	977
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NDF:COUNT?	979
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NNDF:COUNT?	981
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NDF:SEConds?	983
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NNDF:SEConds?	985
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERhead	987
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERhead?	989
:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERhead?	991
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:DISabled	993
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OH:HOP:OVERwrite:ENABLEd	994
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OH:HOP:OVERwrite:ENABLEd?	997
:FETCH[1..n]:DATA:TELEcom:SDHSonet: OH:HOP:OVERwrite:STATus?	1000
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH: LOP:OVERhead	1002
:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH: LOP:OVERhead?	1004
:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH: LOP:OVERhead?	1006
:FETCH[1..n]:DATA:TELEcom:SDHSonet:HOP: PM:STATistics?	1008
:FETCH[1..n]:DATA:TELEcom:SDHSonet:LOP: PM:STATistics?	1011

Contents

SDH Command Reference	1014
:SOURCE[1..n]:DATA:TELEcom:SDH:OVERhead: J:ENABled	1014
:SOURCE[1..n]:DATA:TELEcom:SDH:OVERhead: J:ENABled?	1015
:SOURCE[1..n]:DATA:TELEcom:SDH:OVERhead :J:OVERwrite:ENABled	1016
:SOURCE[1..n]:DATA:TELEcom:SDH:OVERhead :J:OVERwrite:ENABled?	1018
:SOURCE[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATTern	1020
:SOURCE[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATTern?	1022
:SOURCE[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATTern:B16	1024
:SOURCE[1..n]:DATA:TELEcom:SDH:OVERhead:J:PATTern:B16?	1025
:SOURCE[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATTern:B64	1027
:SOURCE[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATTern:B64?	1029
:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead: J:TIM	1031
:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM?	1032
:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATTern	1033
:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:TIM:PATTern?	1035
:FETCh[1..n]:DATA:TELEcom:SDH:OVERhead:J:TIM:PATTern:RECeived?	1037
:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:TIM:PATTern:B16	1039
:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATTern:B16?	1040
:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:TIM:PATTern:B64	1042
:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:TIM:PATTern:B64?	1044
:SOURCE[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABel	1046
:SOURCE[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABel?	1049
:SENSe[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABel:EXPeCted	1052
:SENSe[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABel:EXPeCted?	1055
:FETCh[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABel?	1058
:SENSe[1..n]:DATA:TELEcom:SDH:HOP:PUNeq	1059
:SENSe[1..n]:DATA:TELEcom:SDH:HOP:PUNeq?	1060
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:TYPE	1061
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:TYPE?	1064
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH	1066
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH?	1067
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:MANual:TYPE	1069
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:MANual:TYPE?	1071
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AMOUNT	1073
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AMOUNT?	1075
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:INJect	1077
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:TYPE	1078
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:TYPE?	1080
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:RATE	1082
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:RATE?	1084
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated	1086
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated?	1088

:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:CONTInuous	1090
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:CONTInuous?	1092
:SOURCE[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel	1094
:SOURCE[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel?	1096
:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel:EXPeCted	1098
:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel:EXPeCted?	1100
:FETCh[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel?	1102
:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PUNeq	1104
:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PUNeq?	1105
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:TYPE	1106
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:TYPE?	1109
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH	1111
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH?	1113
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:MANual:TYPE	1115
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:MANual:TYPE?	1117
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AMOUnt	1119
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AMOUnt?	1121
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:INJect	1123
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated:TYPE	1124
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated:TYPE?	1126
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated:RATE	1128
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated:RATE?	1130
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated	1132
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated?	1134
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated:CONTInuous	1136
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated:CONTInuous?	1138
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:HISTory?	1140
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:SEConds?	1142
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:CURRent?	1144
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:COUNT?	1146
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:RATE?	1148
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:PATH:HISTory?	1150
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:PATH:SEConds?	1154
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:PATH:CURRent?	1158
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:HISTory?	1162
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:SEConds?	1164
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:CURRent?	1166
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:COUNT?	1168
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:RATE?	1170
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP: PATH:HISTory?	1172
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP: PATH:SEConds?	1176
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP: PATH:CURRent?	1180

Contents

:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:TYPE	1184
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:TYPE?	1186
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:MODE	1188
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:MODE?	1190
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:DURation	1192
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:DURation?	1195
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:PERiod	1197
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:PERiod?	1200
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt	1202
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt?	1204
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:TYPE	1206
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:TYPE?	1209
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:MODE	1211
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:MODE?	1213
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:DURation	1215
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:DURation?	1218
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:PERiod	1220
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:PERiod?	1223
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt	1225
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt?	1227
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:TYPE	1229
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:TYPE?	1231
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:MODE	1233
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:MODE?	1235
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:DURation	1237
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:DURation?	1240
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:PERiod	1242
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:PERiod?	1245
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt	1247
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt?	1249
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:TYPE	1251
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:TYPE?	1254
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:MODE	1257
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:MODE?	1259
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:DURation	1261
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:DURation?	1264
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:PERiod	1266
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:PERiod?	1269
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt	1271
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt?	1273
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: TYPE	1275
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: TYPE?	1277

:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS	1279
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS?	1280
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: MANual:TYPE	1281
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: MANual:TYPE?	1282
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: AMOunt	1284
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: AMOunt?	1286
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: INJect	1288
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:TYPE	1289
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:TYPE?	1290
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:RATE	1292
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:RATE?	1294
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated	1296
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated?	1298
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:CONTInuous	1300
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:CONTInuous?	1302
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS:TYPE	1304
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS:TYPE?	1306
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS	1308
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: MS?	1309
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:MS: MANual:TYPE	1310
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:MS: MANual:TYPE?	1311
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:MS: AMOunt	1313
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:MS: AMOunt?	1315
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:MS: INJect	1317
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS: HISTory?	1318
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS: SEConds?	1320
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS: CURRent?	1322
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS: HISTory?	1324
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS: SEConds?	1326
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS: CURRent?	1328
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS: COUNT?	1330
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS: RATE?	1332
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS: HISTory?	1334
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS: SEConds?	1336
:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS: CURRent?	1338
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS: HISTory?	1340
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS: SEConds?	1342
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS: CURRent?	1344
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS: COUNT?	1346
:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS: RATE?	1348
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:TYPE	1350
:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:TYPE?	1352

Contents

:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: AUTomated:RATE	1354
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: AUTomated:RATE?	1356
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: AUTomated	1358
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: AUTomated?	1360
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: AUTomated:CONTInuous	1362
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: AUTomated:CONTInuous?	1364
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR: RS: BURSt:TYPE	1366
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR: RS: BURSt:TYPE?	1368
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:RS: BURSt:MODE	1370
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:RS: BURSt:MODE?	1372
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:RS: BURSt:DURation	1374
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:RS: BURSt:DURation?	1376
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:RS: BURSt:PERiod	1378
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:RS: BURSt:PERiod?	1381
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:RS: BURSt	1383
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:RS: BURSt?	1385
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE	1387
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE?	1389
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:MODE	1391
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:MODE?	1393
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:DURation	1395
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:DURation?	1397
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:PERiod	1399
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:PERiod?	1402
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt	1404
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt?	1406
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: BURSt:TYPE	1408
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: BURSt:TYPE?	1410
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: BURSt:MODE	1412
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: BURSt:MODE?	1414
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: BURSt:DURation	1416
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: BURSt:DURation?	1419
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: BURSt:PERiod	1421
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: BURSt:PERiod?	1424
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: BURSt	1426
:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:MS: BURSt?	1428
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:TYPE	1430
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:TYPE?	1432
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:MODE	1434
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:MODE?	1436
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:DURation	1438
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:DURation?	1440

:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS:BURSt:PERiod	1442
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS:BURSt:PERiod?	1444
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS:BURSt	1446
:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS:BURSt?	1448
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]	1450
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]?	1453
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: DISabled	1456
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]:ENABled	1457
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]:ENABled?	1460
:FETCh[1..n]:DATA:TELEcom:SDH:OH:RS: OVERwrite:STATus?	1463
:SENSe[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]?	1464
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]	1467
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]?	1471
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS: DISabled	1475
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS: OVERwrite[1..n]:ENABled	1476
:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS: OVERwrite[1..n]:ENABled?	1480
:FETCh[1..n]:DATA:TELEcom:SDH:OH:MS: OVERwrite:STATus?	1484
:SENSe[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]?	1485
:FETCh[1..n]:DATA:TELEcom:SDH:RS:PM: STATistics?	1489
:FETCh[1..n]:DATA:TELEcom:SDH:MS:PM: STATistics?	1492

Contents

OTN Command Reference	1495
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: FEC	1495
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: FEC?	1497
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: FEC	1499
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: FEC?	1501
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SCRambler	1503
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SCRambler?	1504
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SCRambler	1506
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SCRambler?	1507
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:SAPI:B	1509
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:SAPI:B?	1510
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:DAPI:B	1511
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:DAPI:B?	1512
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:OPSPec:B	1513
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:OPSPec:B?	1514
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:SAPI:B	1515
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:SAPI:B?	1516
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:DAPI:B	1517
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:DAPI:B?	1518
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:OPSPec:B	1520
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:OPSPec:B?	1521
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:SAPI:B16	1523
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:SAPI:B16?	1524
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:DAPI:B16	1526
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:DAPI:B16?	1527
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OPSPec:B32	1529
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OPSPec:B32?	1530
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:SAPI:B	1531
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:SAPI:B?	1532
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:DAPI:B	1533
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:DAPI:B?	1534
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:OPSPec:B	1535
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:OPSPec:B?	1536
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:PM:SAPI:B	1537
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:PM:SAPI:B?	1538
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:PM:DAPI:B	1540
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:PM:DAPI:B?	1541
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:PM:OPSPec:B	1543
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:PM:OPSPec:B?	1544
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OVERwrite:ENABLED	1545
:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OVERwrite:ENABLED?	1546
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:SAPI:EXPEcted	1547

:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:SAPI:EXPeCted?	1549
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:DAPI:EXPeCted	1551
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:DAPI:EXPeCted?	1553
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:SAPI:EXPeCted	1555
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:SAPI:EXPeCted?	1556
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:DAPI:EXPeCted	1558
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:DAPI:EXPeCted?	1559
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:SAPI:EXPeCted	1561
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:SAPI:EXPeCted?	1562
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:DAPI:EXPeCted	1564
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:DAPI:EXPeCted?	1566
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTISAPI:EXPeCted	1568
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:EXPeCted?	1569
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTIDAPI:EXPeCted	1571
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:EXPeCted?	1572
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:SAPI:B?	1574
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:DAPI:B?	1575
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:OPSPec:B?	1576
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:SAPI:EXPeCted	1577
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:SAPI:EXPeCted?	1578
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:DAPI:EXPeCted	1580
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:DAPI:EXPeCted?	1581
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:SAPI:B?	1583
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:DAPI:B?	1584
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:OPSPec:B?	1585
:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:SAPI:B16?	1586
:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:B16?	1587
:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:OPSPec:B32?	1588
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:SAPI:B?	1589
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:DAPI:B?	1590
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:OPSPec:B?	1591
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:SAPI:B?	1592
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:DAPI:B?	1593
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:OPSPec:B?	1594
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:TIM	1595
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:TIM?	1597
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:TIM	1599
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:TIM?	1601
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TTI:TIM	1603
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TTI:TIM?	1605
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:TTI:TIM	1607
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:TTI:TIM?	1609

Contents

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:TIM	1611
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:TIM?	1613
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: CONFig:TCM[1..n]	1615
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: CONFig:TCM[1..n]?	1616
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:CONFig:TCM[1..n]	1618
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:CONFig:TCM[1..n]?	1619
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TTI:OVERwrite:ENABLEd	1621
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TTI:OVERwrite:ENABLEd?	1622
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:INDIcation	1624
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:INDIcation?	1626
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:CODE	1628
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:CODE?	1630
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:IDENtifier	1632
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:IDENtifier?	1634
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OPSPec	1636
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OPSPec?	1638
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:INDIcation?	1640
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:CODE?	1642
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:IDENtifier?	1644
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:SPECific?	1646
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OVERwrite:ENABLEd	1648
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OVERwrite:ENABLEd?	1649
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPe	1651
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPe?	1655
:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPe:RECeived?	1659
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPe	1662
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPe?	1667
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPe:OVERwrite:ENABLEd	1671
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPe:OVERwrite:ENABLEd?	1673
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE	1674
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE?	1676
:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE:RECeived?	1678
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE	1679
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE?	1681
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PLM	1683
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PLM?	1685
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:TYPE	1687
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:TYPE?	1689
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]	1691
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]?	1693
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:MANual:TYPE	1695
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:MANual:TYPE?	1697

:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AMOUNT	1699
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AMOUNT?	1701
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:INJect	1703
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:TYPE	1704
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:TYPE?	1706
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:RATE	1708
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:RATE?	1710
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated	1712
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated?	1714
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:CONTInuous	1716
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:CONTInuous?	1718
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE	1720
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE?	1722
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]	1724
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]?	1726
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TYPE	1728
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TYPE?	1731
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]	1733
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]?	1735
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:MANual:TYPE	1737
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:MANual:TYPE?	1739
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:MANual:TCM[1..n]:TYPE	1741
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:MANual:TCM[1..n]:TYPE?	1743
SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:AMOUNT	1745
SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:AMOUNT?	1747
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:MANual:TCM[1..n]:TYPE	1749
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:MANual:TCM[1..n]:TYPE?	1751
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:AMOUNT	1753
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:AMOUNT?	1755
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AMOUNT?	1757
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AMOUNT?	1759
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:INJect	1761
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:INJect	1762
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:INJect	1763
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TYPE	1764
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TYPE?	1766
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:RATE	1768
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:RATE?	1770
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated	1772
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated?	1774
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:CONTInuous	1776
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:CONTInuous?	1778

Contents

:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:MANual:TYPE	1780
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:MANual:TYPE?	1782
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AMOUnt	1784
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AMOUnt?	1786
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:INJect	1788
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated:TYPE	1789
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated:TYPE?	1791
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated:RATE	1794
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated:RATE?	1796
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated	1798
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated?	1800
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated:CONTInuous	1802
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated:CONTInuous? ...	1804
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE	1806
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE?	1807
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]	1808
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]?	1809
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TCM[1..n]:TYPE	1810
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TCM[1..n]:TYPE?	1812
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODUTCM[1..n]	1814
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODUTCM[1..n]?	1816
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:TYPE	1818
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:TYPE?	1820
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: E:ODUTCM[1..n]	1822
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: E:ODUTCM[1..n]?	1824
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:HISTory?	1826
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:SEConds?	1829
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:CURRent?	1832
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:HISTory?	1835
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:SEConds?	1837
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:CURRent?	1839
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:COUNT?	1841
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:RATE?	1843
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:HISTory?	1845
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:SEConds?	1849
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:CURRent?	1852
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:HISTory?	1855
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:SEConds?	1857
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:CURRent?	1859
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:COUNT?	1861
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:RATE?	1863
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TCM[1..n]:HISTory?	1865

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TCM[1..n]:SEConDs?	1868
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TCM[1..n]:CURRent?	1871
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:HISToRy?	1874
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:SEConDs?	1876
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:CURRent?	1878
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:COUnT?	1880
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:RATE?	1882
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:HISToRy?	1884
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:SEConDs?	1887
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:CURRent?	1890
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:HISToRy?	1893
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:SEConDs?	1895
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:CURRent?	1897
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:COUnT?	1899
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:RATE?	1901
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TCM[1..n]:TYPE	1903
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TCM[1..n]:TYPE?	1905
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TCM[1..n]:TYPE	1907
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TCM[1..n]:TYPE?	1909
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TCM[1..n]:RATE	1911
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TCM[1..n]:RATE?	1913
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TCM[1..n]:RATE	1915
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TCM[1..n]:RATE?	1917
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:AUTomated	1919
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:AUTomated?	1921
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:AUTomated	1923
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:AUTomated?	1925
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous	1927
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TCM[1..n]: CONTInuous?	1929
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TCM[1..n]: CONTInuous	1931
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TCM[1..n]: CONTInu-	

Contents

ous?	1933
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TCM[1..n]:RATE	1935
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TCM[1..n]:RATE?	1937
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:HISTory?	1939
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:SEConds?	1941
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:CURRent?	1943
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:HISTory?	1945
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:SEConds?	1948
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:CURRent?	1951
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:COUNt?	1954
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:RATE?	1957
:SOURCE[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:OVERhead	1960
:SOURCE[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:OVERhead?	1963
:SOURCE[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:DEFault	1966
:SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]:OVERhead?	1967
:SOURCE[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead	1970
:SOURCE[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead?	1974
:SOURCE[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:DEFault	1979
:SENSe[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead?	1980
:SOURCE[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:OVERhead	1984
:SOURCE[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:OVERhead?	1988
:SOURCE[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:DEFault	1993
:SENSe[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:OVERhead?	1994
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:FEC	1999
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:FEC?	2000
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: FEC	2001
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: FEC?	2002
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SCRambler	2003
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SCRambler?	2004
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SCRambler	2005
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SCRambler?	2006
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SM:SAPI:B16	2007
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SM:SAPI:B16?	2008
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SM:DAPI:B16	2010
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SM:DAPI:B16?	2011
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SM:OPSPec:B32	2013
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SM:OPSPec:B32?	2014
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SM:OVERwrite:ENABLEd	2016
:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SM:OVERwrite:ENABLEd?	2017
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:SAPI:EXPEcted	2018
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:SAPI:EXPEcted?	2019
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:DAPI:EXPEcted	2021

:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:DAPI:EXPeCted?	2022
:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:SAPI:B16?	2024
:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:DAPI:B16?	2025
:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:OPSPeC:B32?	2026
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:TIM	2027
:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:TIM?	2029
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:OVERwrite:ENABled	2031
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:OVERwrite:ENABled?	2032
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:INDIcation	2034
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:INDIcation?	2037
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:CODE	2039
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:CODE?	2041
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:IDENTifier	2043
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:IDENTifier?	2045
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:OPSPeC	2047
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:OPSPeC?	2049
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:INDIcation?	2051
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:CODE?	2053
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:IDENTifier?	2055
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:SPECIfic?	2057
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:OVERwrite:ENABled	2059
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:OVERwrite:ENABled?	2060
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPe	2062
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPe?	2065
:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E: PTYPe:RECeived?	2068
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPe	2071
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPe?	2074
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPe:OVERwrite:ENABled	2077
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPe:OVERwrite:ENABled?	2079
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PCODE	2081
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PCODE?	2082
:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PCODE:RECeived?	2084
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E: PCODE	2085
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PCODE?	2086
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E: PLM	2088
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PLM?	2089
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: MSIM	2091
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: MSIM?	2092
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:TYPE	2093
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:TYPE?	2096
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E	2098
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E?	2100

Contents

:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:MANual:TYPE	2102
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:MANual:TYPE?	2104
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AMOUnt	2106
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AMOUnt?	2108
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:INJect	2110
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:TYPE	2111
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:TYPE?	2114
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:RATE	2116
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:RATE?	2118
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated	2120
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated?	2122
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:CONTInuous	2124
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:CONTInuous?	2126
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:TYPE	2128
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:TYPE?	2131
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:RATE	2133
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:RATE?	2135
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated	2137
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated?	2139
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:CONTInuous ..	2141
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:CONTInuous? ..	2143
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:TYPE	2145
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:TYPE?	2147
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E	2149
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E?	2151
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TYPE	2153
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TYPE?	2156
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E	2158
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E?	2160
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:MANual:TYPE	2162
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:MANual:TYPE?	2164
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AMOUnt	2166
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AMOUnt?	2168
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:INJect	2170
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:MANual:TYPE	2171
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:MANual:TYPE?	2173
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AMOUnt	2175
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AMOUnt?	2177
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:INJect	2179
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:HISTory?	2180
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:SEConds?	2183
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:CURRent?	2186

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:HISTory?	2189
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:SEConds?	2191
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:CURRent?	2193
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:COUnT?	2195
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:RATE?	2197
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:HISTory?	2199
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:SEConds?	2203
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:CURRent?	2207
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:HISTory?	2211
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:SEConds?	2213
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:CURRent?	2215
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:COUnT?	2217
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:RATE?	2219
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TYPE	2221
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TYPE?	2224
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:RATE	2226
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:RATE?	2228
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated	2230
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated?	2232
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:CONTInuous	2234
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:CONTInuous?	2236
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:HISTory?	2238
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:SEConds?	2240
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:CURRent?	2242
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:HISTory?	2244
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:SEConds?	2247
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:CURRent?	2250
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:COUnT?	2253
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:RATE?	2256
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:OVERhead	2259
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:OVERhead?	2262
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:DEFault	2265
:SENSe[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:OVERhead?	2266
:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:E:OVERhead	2269
:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:E:OVERhead?	2273
:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:E:DEFault	2278
:SENSe[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:E:OVERhead?	2279
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:E:OVERhead	2283
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:E:OVERhead?	2287
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:E:DEFault	2292
:SENSe[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:E:OVERhead?	2293

Contents

DSn Command Reference	2298
:SOURce[1..n]:DATA:TELEcom:DS[1..n]: PAYLoad:FRAMing	2298
:SOURce[1..n]:DATA:TELEcom:DS[1..n]: PAYLoad:FRAMing?	2300
:SENSe[1..n]:DATA:TELEcom:DS[1..n]: PAYLoad:FRAMing	2302
:SENSe[1..n]:DATA:TELEcom:DS[1..n]: PAYLoad:FRAMing?	2304
:SOURce[1..n]:DATA:TELEcom:DS:ENABled	2306
:SOURce[1..n]:DATA:TELEcom:DS:ENABled?	2307
:SENSe[1..n]:DATA:TELEcom:DS:ENABled	2308
:SENSe[1..n]:DATA:TELEcom:DS:ENABled?	2309
:SOURce[1..n]:DATA:TELEcom:DS:MODE	2310
:SOURce[1..n]:DATA:TELEcom:DS:MODE?	2312
:SENSe[1..n]:DATA:TELEcom:DS:MODE	2314
:SENSe[1..n]:DATA:TELEcom:DS:MODE?	2316
:SOURce[1..n]:DATA:TELEcom:DS:ZCS	2318
:SOURce[1..n]:DATA:TELEcom:DS:ZCS?	2320
:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: IDLE	2322
:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: IDLE?	2323
:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: TONE	2325
:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: TONE?	2327
:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTEnt	2329
:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTEnt?	2331
:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTEnt	2333
:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTEnt?	2335
:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTEnt:TYPE	2337
:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTEnt:TYPE?	2339
:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTEnt:TYPE	2341
:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTEnt:TYPE?	2343
:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTEnt:ALL	2345
:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTEnt:ALL	2347
:SOURce[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]:TYPE	2348
:SOURce[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]:TYPE?	2350
:SOURce[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]	2352
:SOURce[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]?	2354
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:MANual:TYPE	2356
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:MANual:TYPE?	2359
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AMOUnt	2361
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AMOUnt?	2363
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:INJect	2365
:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:HISTory?	2366
:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:SECOnds?	2369
:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:CURRent?	2372
:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:COUNt?	2375

:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:RATE?	2378
:FETCh[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]:HISTory?	2381
:FETCh[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]:SEConds?	2384
:FETCh[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]:CURRent?	2387
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:TYPE	2390
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:TYPE?	2393
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:RATE	2395
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:RATE?	2397
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated	2399
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated?	2401
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:CONTInuous	2403
:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:CONTInuous?	2405
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC	2407
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC?	2408
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:CODeword	2409
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:CODeword?	2413
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:AMOUnt	2416
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:AMOUnt?	2418
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:INJect	2420
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous:CODeword	2421
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous:CODeword?	2425
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous	2428
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous?	2430
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:CODeword	2432
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:CODeword?	2434
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:AMOUnt	2436
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:AMOUnt?	2438
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:CODeword	2440
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:CODeword?	2443
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:AMOUnt	2446
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:AMOUnt?	2448
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:INJect	2450
:SENSe[1..n]:DATA:TELEcom:DSN:FEAC	2451
:SENSe[1..n]:DATA:TELEcom:DSN:FEAC?	2452
:FETCh[1..n]:DATA:TELEcom:DSN:FEAC: MESSAge?	2453
:FETCh[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol?	2455
:FETCh[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel?	2457
:FETCh[1..n]:DATA:TELEcom:DSN:FEAC: LINK?	2459
:FETCh[1..n]:DATA:TELEcom:DSN:DS[1..n]:PM:STATistics?	2461

Contents

PDH Command Reference	2464
:SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled	2464
:SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled?	2465
:SENSe[1..n]:DATA:TELEcom:PDH:E:ENABled	2466
:SENSe[1..n]:DATA:TELEcom:PDH:E:ENABled?	2467
:SOURce[1..n]:DATA:TELEcom:PDH:E:MODE	2468
:SOURce[1..n]:DATA:TELEcom:PDH:E:MODE?	2470
:SENSe[1..n]:DATA:TELEcom:PDH:E:MODE	2472
:SENSe[1..n]:DATA:TELEcom:PDH:E:MODE?	2474
:SOURce[1..n]:DATA:TELEcom:PDH:E:ZCS	2476
:SOURce[1..n]:DATA:TELEcom:PDH:E:ZCS?	2477
:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: IDLE	2479
:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: IDLE?	2480
:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:TONE	2482
:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:TONE?	2484
:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:CONTent	2486
:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:CONTent?	2488
:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent	2490
:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent?	2492
:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:CONTent:TYPE	2494
:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:CONTent:TYPE?	2496
:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE	2498
:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE?	2500
:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:CONTent:ALL	2502
:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:ALL	2503
:SOURce[1..n]:DATA:TELEcom:PDH:E[1..n]: FRAMing	2504
:SOURce[1..n]:DATA:TELEcom:PDH:E[1..n]: FRAMing?	2506
:SENSe[1..n]:DATA:TELEcom:PDH:E[1..n]: FRAMing	2508
:SENSe[1..n]:DATA:TELEcom:PDH:E[1..n]: FRAMing?	2510
:SOURce[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:TYPE	2512
:SOURce[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:TYPE?	2514
:SOURce[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]	2516
:SOURce[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]?	2518
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:MANual:TYPE	2520
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:MANual:TYPE?	2522
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AMOUnt	2524
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AMOUnt?	2526
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:INJect	2528
:FETCh[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:HISTory?	2529
:FETCh[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:SEConds?	2532
:FETCh[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:CURRent?	2534
:FETCh[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:COUNt?	2537

:FETCh[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:RATE?	2539
:FETCh[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:HISTory?	2541
:FETCh[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:SEConds?	2544
:FETCh[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:CURRent?	2547
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:TYPE	2550
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:TYPE?	2552
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:RATE	2554
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:RATE?	2556
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated	2558
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated?	2560
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:CONTInuous	2562
:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:CONTInuous?	2564
:FETCh[1..n]:DATA:TELEcom:PDH:E[1..n]:PM: STATistics?	2566

Contents

SDT and RTD Command Reference	2569
:SENSe[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE	2569
:SENSe[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE?	2572
:SENSe[1..n]:DATA:TELEcom:SDT:SONet: DSELection	2575
:SENSe[1..n]:DATA:TELEcom:SDT:SONet: DSELection?	2579
:SENSe[1..n]:DATA:TELEcom:SDT:SDH:LAYer: TYPE	2583
:SENSe[1..n]:DATA:TELEcom:SDT:SDH:LAYer: TYPE?	2585
:SENSe[1..n]:DATA:TELEcom:SDT:SDH: DSELection	2587
:SENSe[1..n]:DATA:TELEcom:SDT:SDH: DSELection?	2592
:SENSe[1..n]:DATA:TELEcom:SDT:PDH:LAYer: TYPE	2596
:SENSe[1..n]:DATA:TELEcom:SDT:PDH:LAYer: TYPE?	2598
:SENSe[1..n]:DATA:TELEcom:SDT:PDH: DSELection	2600
:SENSe[1..n]:DATA:TELEcom:SDT:PDH: DSELection?	2602
:SENSe[1..n]:DATA:TELEcom:SDT:DSN:LAYer: TYPE	2605
:SENSe[1..n]:DATA:TELEcom:SDT:DSN:LAYer: TYPE?	2607
:SENSe[1..n]:DATA:TELEcom:SDT:DSN: DSELection	2609
:SENSe[1..n]:DATA:TELEcom:SDT:DSN: DSELection?	2611
:SENSe[1..n]:DATA:TELEcom:SDT:NDTime	2613
:SENSe[1..n]:DATA:TELEcom:SDT:NDTime?	2615
:SENSe[1..n]:DATA:TELEcom:SDT:TPERiod	2617
:SENSe[1..n]:DATA:TELEcom:SDT:TPERiod?	2619
:SENSe[1..n]:DATA:TELEcom:SDT	2621
:SENSe[1..n]:DATA:TELEcom:SDT?	2622
:FETCh[1..n]:DATA:TELEcom:SDT:SHORtest?	2623
:FETCh[1..n]:DATA:TELEcom:SDT:LONGest?	2625
:FETCh[1..n]:DATA:TELEcom:SDT:LAST?	2627
:FETCh[1..n]:DATA:TELEcom:SDT:AVERage?	2629
:FETCh[1..n]:DATA:TELEcom:SDT:TOTal?	2631
:FETCh[1..n]:DATA:TELEcom:SDT:ALARm: HISTory?	2633
:FETCh[1..n]:DATA:TELEcom:SDT:ALARm: CURRent?	2635
:FETCh[1..n]:DATA:TELEcom:SDT:ALARm: SEConds?	2637
:FETCh[1..n]:DATA:TELEcom:SDT:COUNt?	2639
:SENSe[1..n]:DATA:TELEcom:RTD:MODE	2641
:SENSe[1..n]:DATA:TELEcom:RTD:MODE?	2642
:SENSe[1..n]:DATA:TELEcom:RTD	2643
:SENSe[1..n]:DATA:TELEcom:RTD?	2644
:FETCh[1..n]:DATA:TELEcom:RTD:DELay: STATus?	2645
:FETCh[1..n]:DATA:TELEcom:RTD:DELay:LAST?	2647
:FETCh[1..n]:DATA:TELEcom:RTD:DELay: MAXimum?	2648
:FETCh[1..n]:DATA:TELEcom:RTD:DELay: MINimum?	2649
:FETCh[1..n]:DATA:TELEcom:RTD:DELay: AVERage?	2650
:FETCh[1..n]:DATA:TELEcom:RTD:COUNt: SUCCEssful?	2651

:FETCh[1..n]:DATA:TELEcom:RTD:COUNT: FAILED?	2652
:SENSe[1..n]:DATA:TELEcom:RTD:RESet	2653
:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer: TYPE	2654
:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer: TYPE?	2657
:SENSe[1..n]:DATA:TELEcom:SDT:OTN: DSElection	2660
:SENSe[1..n]:DATA:TELEcom:SDT:OTN: DSElection?	2666

Contents

EoOTN Command Reference	2671
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:ADDReSS:DEStination	2671
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:ADDReSS:DEStination?	2673
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:ADDReSS:SOURce	2675
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:ADDReSS:SOURce?	2677
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:DATAlink:SIZE	2679
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:DATAlink:SIZE?	2681
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:RATE	2684
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:RATE?	2686
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:VLAN	2688
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:VLAN?	2690
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:VLAN:ID	2692
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:VLAN:ID?	2695
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:VLAN:PRiority	2698
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:VLAN:PRiority?	2701
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:VLAN:TYPE?	2703
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:ENABled	2705
:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:ENABled?	2707
:FETCh[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:HISTory?	2709
:FETCh[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:CURRent?	2711
:FETCh[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:SEConDs?	2713
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:HISTory?	2715
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:CURRent?	2717
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:SEConDs?	2719
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:COUNT?	2721
:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:RATE?	2723
:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:HISTory?	2725
:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:CURRent?	2728
:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:SEConDs?	2731
:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:COUNT?	2734
:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:RATE?	2737
:SOURCE[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical	2739
:SOURCE[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical?	2741
:SOURCE[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:TYPE	2743
:SOURCE[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:TYPE?	2744
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE	2746
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE?	2747
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUnt	2749
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUnt?	2751
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:INJect	2753
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:TYPE	2754
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:TYPE?	2756

:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE	2758
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE?	2760
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated	2762
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated?	2764
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTinuous	2766
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:CONTinuous?	2768
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE	2770
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:MANual:TYPE?	2771
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT	2773
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AMOUNT?	2775
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:INJect	2777
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:TYPE	2778
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:TYPE?	2780
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:RATE	2782
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:RATE?	2784
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated	2786
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated?	2788
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTinuous	2790
:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:PHYSical:AUTomated:CONTinuous?	2792
:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe:BANdwidth?	2794
:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe:COUNt:RX?	2795
:SOURCE[1..n]:DATA:TELEcom:ETHernet:FRAMe:COUNt:TX?	2797
:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe:RATE?	2799
:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe:UTILization?	2800
:SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZe: COUNT?	2801

Contents

Summary Page Command Reference	2804
:FETCh[1..n]:DATA:TELEcom:SUMMary:TEST: HISTory?	2804
:FETCh[1..n]:DATA:TELEcom:SUMMary:TEST: CURRent?	2807
:FETCh[1..n]:DATA:TELEcom:SUMMary:PORT: HISTory?	2809
:FETCh[1..n]:DATA:TELEcom:SUMMary:PORT: CURRent?	2811
:FETCh[1..n]:DATA:TELEcom:SUMMary:OTN: HISTory?	2813
:FETCh[1..n]:DATA:TELEcom:SUMMary:OTN: CURRent?	2815
:FETCh[1..n]:DATA:TELEcom:SUMMary: SONetsdh:HISTory?	2817
:FETCh[1..n]:DATA:TELEcom:SUMMary: SONetsdh:CURRent?	2820
:FETCh[1..n]:DATA:TELEcom:SUMMary: DSNPdh:HISTory?	2823
:FETCh[1..n]:DATA:TELEcom:SUMMary: DSNPdh:CURRent?	2825
:FETCh[1..n]:DATA:TELEcom:SUMMary: PATTern:HISTory?	2827
:FETCh[1..n]:DATA:TELEcom:SUMMary: PATTern:CURRent?	2829
:FETCh[1..n]:DATA:TELEcom:SUMMary:OTHer:HISTory?	2831
:FETCh[1..n]:DATA:TELEcom:SUMMary:OTHer:CURRent?	2833
:FETCh[1..n]:DATA:TELEcom:SUMMary: COFFset:HISTory?	2835
:FETCh[1..n]:DATA:TELEcom:SUMMary: COFFset:CURRent?	2837
:FETCh[1..n]:DATA:TELEcom:SUMMary: COFFset:FREQuency?	2839
:FETCh[1..n]:DATA:TELEcom:SUMMary: COFFset:OFFSet?	2840
Exhaustive Command Reference	2841
:FETCh[1..n]:DATA:TELEcom:ALARm:HISTory?	2841
:FETCh[1..n]:DATA:TELEcom:ALARm:SEConds?	2846
:FETCh[1..n]:DATA:TELEcom:ALARm:CURRent?	2851
:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?	2856
:FETCh[1..n]:DATA:TELEcom:ERRor:SEConds?	2861
:FETCh[1..n]:DATA:TELEcom:ERRor:CURRent?	2866
:FETCh[1..n]:DATA:TELEcom:ERRor:COUnT?	2871
:FETCh[1..n]:DATA:TELEcom:ERRor:RATE?	2876
Logger Command Reference	2881
:FETCh[1..n]:DATA:TELEcom:LOGGer:EVENTs?	2881
:FETCh[1..n]:DATA:TELEcom:LOGGer:LIST?	2882

GFP Commands	2884
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: COUNT:TX?	2884
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: RATE:TX?	2886
:FETCh[1..n]:DATA:TELEcom:GFP:CStat: COUNT:TX?	2888
:FETCh[1..n]:DATA:TELEcom:GFP:CStat: RATE:TX?	2890
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: BANDwidth:TX?	2892
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: EFFiciency:TX?	2893
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: COUNT:RX?	2894
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: RATE:RX?	2897
:FETCh[1..n]:DATA:TELEcom:GFP:CStat: COUNT:RX?	2899
:FETCh[1..n]:DATA:TELEcom:GFP:CStat: RATE:RX?	2901
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: BANDwidth:RX?	2903
:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: EFFiciency:RX?	2904
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: HISTory?	2905
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: CURRent?	2907
:SOURce[1..n]:DATA:TELEcom:GFP: CONFIguration:EXI	2909
:SOURce[1..n]:DATA:TELEcom:GFP: CONFIguration :EXI?	2910
:SOURce[1..n]:DATA:TELEcom:GFP: CONFIguration:UPI?	2911
:SENSe[1..n]:DATA:TELEcom:GFP: CONFIguration:UPI?	2914
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAMe:TYPE	2917
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAMe:TYPE?	2918
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAMe	2919
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAMe?	2920
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:MANual:TYPE	2921
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:MANual:TYPE?	2923
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AMOUnt	2925
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AMOUnt?	2926
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:INJect	2928
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:TYPE	2929
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:TYPE?	2931
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:RATE	2933
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:RATE?	2934
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated	2936
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated?	2937
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:CONTInuous	2938
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:CONTInuous?	2939
:FETCh[1..n]:DATA:TELEcom:GFP:CONFIg: DELTα?	2940
:SENSe[1..n]:DATA:TELEcom:GFP:FRAMe: MISMatch:COUNT?	2941
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: FRAMe:HISTory?	2943
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: FRAMe:CURRent?	2945
:FETCh[1..n]:DATA:ELecom:GFP:ALARm:FRAMe:SEConds?	2947
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:HISTory?	2949

Contents

:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:CURRent?	2951
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:SEConds?	2953
::FETCh[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:COUnT?	2955
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:RATE?	2957
:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:TYPE	2959
:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:TYPE?	2961
:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:CID	2963
:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:CID?	2964
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:TYPE	2966
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:TYPE?	2968
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:PERiod?	2969
::SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:UPI	2971
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:UPI?	2972
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel	2974
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel?	2975
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated:TYPE	2976
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated:TYPE?	2979
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated:RATE	2982
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated:RATE?	2983
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated:CONTInuous	2985
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated:CONTInuous?	2986
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated	2987
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated?	2988
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:MANual:TYPE	2989
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:MANual:TYPE?	2992
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AMOUnt	2993
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AMOUnt?	2994
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:MANual	2996
:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:MANual?	2997
:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:CID	2999
:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:CID?	3000
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:HISTory?	3001
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:CURRent?	3004
:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:SEConds?	3007
:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel: MISMatch:COUnT?	3010
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:HISTory?	3012
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:CURRent?	3015
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:SEConds?	3018
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:COUnT?	3021
:FETCh[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:Rate?	3024
:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI	3027
:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI?	3029

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI	3031
:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI?	3033
:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI	3035
:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI?	3037
:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI	3039
:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI?	3041
:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID	3043
:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:CID?	3045
:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe	3047
:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader:SPARe?	3049
:FETCh[1..n]:DATA:TELEcom:GFP:OH:DFRames?	3051
:FETCh[1..n]:DATA:TELEcom:GFP:OH:MFRames?	3054
:FETCh[1..n]:DATA:TELEcom:GFP:OH:RPTiframes?	3057
:SOURce[1..n]:DATA:TELEcom:GFP:CLlent:PAYLoad:SIZE	3060
:SOURce[1..n]:DATA:TELEcom:GFP:CLlent:PAYLoad:SIZE?	3061
:SOURce[1..n]:DATA:TELEcom:GFP:CLlent:INTerface	3063

Contents

FC Overclocked 1F	3064
SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]: F:FEC	3064
SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]: F:FEC?	3065
SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]: F:FEC	3066
SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]: F:FEC?	3067
SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]: F:SCRambler	3068
SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]: F:SCRambler?	3069
SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]: F:SCRambler	3070
SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]: F:SCRambler?	3071
:SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:F:SM:SAPI:B16	3072
:SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:F:SM:SAPI:B16?	3073
:SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:F:SM:DAPI:B16	3075
:SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:F:SM:DAPI:B16?	3076
:SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: SM:OPSPec:B32	3078
:SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: SM:OPSPec:B32?	3079
:SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:F:SM:OVERwrite:ENABled	3081
:SOURce[1..n]:DATA:TELecom:OTN:OTU[1..n]:F:SM:OVERwrite:ENABled?	3082
:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:SAPI:EXPEcted	3083
:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:SAPI:EXPEcted?	3084
:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:DAPI:EXPEcted	3086
:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:DAPI:EXPEcted?	3087
:FEtCh[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:SAPI:B16?	3089
:FEtCh[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:DAPI:B16?	3090
:FEtCh[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:OPSPec:B32?	3091
:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:TIM	3092
:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:TIM?	3094
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:TTI:OVERwrite:ENABled	3096
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:TTI:OVERwrite:ENABled?	3097
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:INDIcation	3099
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:INDIcation?	3102
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:CODE	3104
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:CODE?	3106
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:IDENtifier	3108
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:IDENtifier?	3110
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:OPSPec	3112
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:OPSPec?	3114
:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:INDIcation?	3116
:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:CODE?	3118
:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:IDENtifier?	3120
:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:SPECific?	3122
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:OVERwrite:ENABled	3124
:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:FTFL:OVERwrite:ENABled?	3125

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE	3127
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE?	3130
:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F: PTYPE:RECeived?	3133
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE	3136
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE?	3139
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE:OVERwrite:ENABLEd	3142
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE:OVERwrite:ENABLEd?	3144
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PCODE	3146
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PCODE?	3147
:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PCODE:RECeived?	3149
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F: PCODE	3150
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PCODE?	3151
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F: PLM	3153
:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PLM?	3154
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F:TYPE	3156
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F:TYPE?	3159
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F	3161
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F?	3163
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:MANual:TYPE	3165
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:MANual:TYPE?	3167
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AMOUnt	3169
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AMOUnt?	3171
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:INJect	3173
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated:TYPE	3174
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated:TYPE?	3177
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated:RATE	3179
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated:RATE?	3181
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated	3183
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated?	3185
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated:CONTInuous	3187
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated:CONTInuous?	3189
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:TYPE	3191
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:TYPE?	3194
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:RATE	3196
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:RATE?	3198
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated	3200
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated?	3202
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:CONTInuous	3204
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:CONTInuous?	3206
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F:TYPE	3208
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F:TYPE?	3210
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F	3212

Contents

:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F	3213
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TYPE	3215
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TYPE?	3218
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F	3221
:SOURCE[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F	3222
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:MANual:TYPE	3224
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:MANual:TYPE?	3226
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AMOUnt	3228
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AMOUnt?	3230
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:INJect	3232
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:MANual:TYPE	3233
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:MANual:TYPE?	3236
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AMOUnt	3238
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AMOUnt?	3240
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:INJect:	3242
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F:HISTory?	3243
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F:SECOnds?	3247
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F:CURREnt?	3251
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:HISTory?	3255
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:SECOnds?	3257
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:CURREnt?	3259
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:COUnt?	3261
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:RATE?	3263
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:HISTory?	3265
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:SECOnds?	3269
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:CURREnt?	3273
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:HISTory?	3277
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:SECOnds?	3279
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:CURREnt?	3281
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:COUnt?	3283
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:RATE?	3285
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTOmated:TYPE	3287
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTOmated:TYPE?	3289
SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTOmated:RATE	3291
SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTOmated:RATE?	3293
SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTOmated	3295
SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTOmated?	3296
SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTOmated:CONTInuous	3298
SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTOmated:CONTInuous?	3300
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F:HISTory?	3302
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F:SECOnds?	3304
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F:CURREnt?	3306

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:HISTory?	3308
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:SEConds?	3311
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:CURRent?	3314
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:COUNt?	3317
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:RATE?	3320
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:F:OVERhead	3323
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:F:OVERhead?	3326
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:F:DEFault	3329
:SENSe[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:F:OVERhead?	3330
:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:OVERhead	3332
:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:OVERhead?	3336
:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:DEFault	3340
:SENSe[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:OVERhead?	3341
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:F:OVERhead	3345
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:F:OVERhead?	3349
:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:F:DEFault	3354
:SENSe[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:F:OVERhead?	3355
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:SAPI:B	3360
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:SAPI:B?	3361
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:DAPI:B	3362
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:DAPI:B?	3363
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:OPSPec:B	3364
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:OPSPec:B?	3365
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:SAPI:B	3366
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:SAPI:B?	3367
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:DAPI:B	3368
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:DAPI:B?	3369
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:OPSPec:B	3370
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:OPSPec:B?	3371
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:SAPI:EXPeCted	3372
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:SAPI:EXPeCted?	3373
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:DAPI:EXPeCted	3375
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:DAPI:EXPeCted?	3376
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TTI:SAPI:EXPeCted	3378
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TTI:SAPI:EXPeCted?	3379
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TTI:DAPI:EXPeCted	3381
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TTI:DAPI:EXPeCted?	3382
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TTI:SAPI:B?	3384
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TTI:DAPI:B?	3385
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TTI:OPSPec:B?	3386
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:SAPI:B?	3387
:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:DAPI:B?	3388

Contents

:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:OPSPec:B?	3389
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:TIM	3390
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:TIM?	3392
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:TIM	3394
:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:TIM?	3396
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:CONFig:TCM[1..n]	3398
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:CONFig:TCM[1..n]?	3399
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TCM[1..n]:TYPE	3401
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TCM[1..n]:TYPE?	3403
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:F: ODUTCM[1..n]	3405
:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:F: ODUTCM[1..n]?	3407
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TCM[1..n]:HISTory?	3409
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TCM[1..n]:SEConds?	3412
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TCM[1..n]:CURRent?	3415
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:HISTory?	3418
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:SEConds?	3420
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:CURRent?	3422
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:COUNT?	3424
:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:RATE?	3426
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TCM[1..n]:TYPE	3428
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TCM[1..n]:TYPE?	3430
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TCM[1..n]:RATE	3432
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TCM[1..n]:RATE?	3434
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:AUTomated	3436
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:AUTomated?	3438
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU [1..n]:F:AUTomated:TCM[1..n]:CONTInuous?	3440
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU [1..n]:F:AUTomated:TCM[1..n]:CONTInuous?	

3442

:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:MANual:TCM[1..n]:TYPE	3444
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:MANual:TCM[1..n]:TYPE?	3446
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:AMOUNT	3448
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:AMOUNT?	3450
:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:INJect	3452
:SOURCE[1..n]:DATA:TELEcom:OPTical:PORT: WAVElength	3453
:SOURCE[1..n]:DATA:TELEcom:OPTical:PORT: WAVElength?	3454
:SOURCE[1..n]:DATA:TELEcom:OPTical:PORT: IPOLarity	3455
:SOURCE[1..n]:DATA:TELEcom:OPTical:PORT: IPOLarity?	3456
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:SMODE	3457
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:SMODE?	3458
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:OVERwrite	3459
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:OVERwrite?	3460
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:REQuest	3461
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:REQuest?	3464
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:REQuest	3467
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:REQuest?	3470
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:CHANnel	3473
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:CHANnel?	3474
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:PCHannel	3475
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:PCHannel?	3476
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:ARCHitecture	3477
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:ARCHitecture?	3478
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:OMODE	3479
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:OMODE?	3482
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:OMODE	3484
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:OMODE?	3486
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:OMODE	3488
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:OMODE?	3491
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:DNODE	3492
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:DNODE?	3493
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:SNODE	3494
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:SNODE?	3495
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:BREQuest	3496
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:BREQuest?	3497
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:H:SSBits	3498
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:H:SSBits?	3499
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:H:OVERwrite	3500
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:H:OVERwrite?	3501
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:OVERwrite	3502
:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:OVERwrite?	3503

Contents

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:VALue	3504
:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:VALue?	3505
:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:SSMessage	3506
:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:SSMessage?	3509
:SENSe[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:SMODE	3512
:SENSe[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:SMODE?	3513
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:REQuest?	3514
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:REQuest?	3515
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:CHANnel?	3516
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:PCHannel?	3517
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:ARCHitecture?	3518
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:OMODE?	3519
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:OMODE?	3521
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:DNODE?	3523
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:SNODE?	3524
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:BREQuest?	3525
:SENSe[1..n]:DATA:TELEcom:SDHSonet: ADVanced:CAPTure:TRANSitions	3526
:SENSe[1..n]:DATA:TELEcom:SDHSonet: ADVanced:CAPTure:TRANSitions?	3527
:SENSe[1..n]:DATA:TELEcom:SDHSonet: ADVanced:CAPTure	3528
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:H:SSBits?	3529
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:VALue?	3530
:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:SSMessage?	3531
Test Preferences Commands	3534
:OUTPut[1..n]:TELEcom:SDHSonet:PREFErences:DPReferences:LASer:STATus	3534
:OUTPut[1..n]:TELEcom:SDHSonet: PREFErences:DPReferences:LASer:STATus?	3535
:OUTPut[1..n]:TELEcom:SDHSonet:PREFErences:DPReferences:BFOVerwrite:STATus	3536
:OUTPut[1..n]:TELEcom:SDHSonet:PREFErences:DPReferences:BFOVerwrite:STATus?	3537
:OUTPut[1..n]:TELEcom:SDHSonet:PREFErences:DPReferences:SMLTest:PATtern:TYPE	3539
:OUTPut[1..n]:TELEcom:SDHSonet: PREFErences:DPReferences:SMLTest:PATtern: TYPE?	3542
:OUTPut[1..n]:TELEcom:SDHSonet: PREFErences:DPReferences:SMLTest:INVert: STATus	3545
:OUTPut[1..n]:TELEcom:SDHSonet: PREFErences:DPReferences:SMLTest:INVert: STATus?	3546
:OUTPut[1..n]:TELEcom:SDHSonet: PREFErences:DPReferences:SMLTest: RXLTraffic:STATus	3549
:OUTPut[1..n]:TELEcom:SDHSonet: PREFErences:DPReferences:SMLTest: RXLTraffic:STATus?	

3550

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:BACKground: OTNFlex 3552

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:BACKground: OTNFlex? 3554

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:BACKground:OTN3556

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:BACKground:OTN?3558

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:BACKground:HOP3560

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:BACKground: HOP?3562

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:BACKground:LOP3564

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:BACKground:LOP?3566

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:BACKground: DSNPdh 3568

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:BACKground: DSNPdh? 3570

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:LCAS:SOURce: STATus .3572

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:LCAS:SOURce: STATus? 3574

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:OCSTm: COMPUTation:
METHod3576

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:OCSTm: COMPUTation:METH-
od?3578

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:DSN:LCODE: CONFigura-
tion:NAME3580

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:DSN:LCODE: CONFigura-
tion:NAME?3582

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:DSN:LCODE: CONFigura-
tion:LUP3584

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:DSN:LCODE: CONFigura-
tion:LUP?3586

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:DSN:LCODE: CONFigura-
tion:LDOWN3588

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:DSN:LCODE: CONFigura-
tion:LDOWN?3590

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:DSN:LCODE: CONFigura-
tion:EXPort3592

:OUTPut[1..n]:TELEcom:SDHSONet: PREFERences:DPREFERences:DSN:LCODE: CONFigura-

Contents

tion:IMPort	3593
Client Frequency Offset	3594
:SOURce[1..n]:DATA:TELEcom:ETHernet: COFFset:FREQuency:OFFSet	3594
SOURce[1..n]:DATA:TELEcom:ETHernet: COFFset:FREQuency:OFFSet?	3595
:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency?	3597
:SOURce[1..n]:DATA:TELEcom:ETHernet: COFFset:FREQuency:ENABle	3599
:SOURce[1..n]:DATA:TELEcom:ETHernet: COFFset:FREQuency:ENABle?	3600
:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:HISTory?	3602
:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:CURRent?	3604
:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:ALARm:SEConds?	3606
:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency?	3608
:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFIguration:FOANalysis:ENABle	3611
:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFIguration:FOANalysis:ENABle?	3612
:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:CONFIguration:EFREquency?	3613
Multi Channel SDT	3615
:SENSe[1..n]:DATA:TELEcom:SDT:MONItor: CONFIg:THREshold:ENABle	3615
:SENSe[1..n]:DATA:TELEcom:SDT:MONItor: CONFIg:THREshold:ENABle?	3617
:SENSe[1..n]:DATA:TELEcom:SDT:MONItor: CONFIg:THREshold	3619
:SENSe[1..n]:DATA:TELEcom:SDT:MONItor: CONFIg:THREshold?	3620
:SOURce[1..n]:DATA:TELEcom:SDT:CHANnel: SElect	3621
:SOURce[1..n]:DATA:TELEcom:SDT:CHANnel: SElect?	3623
:SOURce[1..n]:DATA:TELEcom:SDT:CHANnel: ENABle:STATus?	3625
:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:LIST?	3626
:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: CATHreshold?	3627
:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: CMONItored?	3628
:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: LODisruption:DURation?	3629
:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: LODisruption:CHANnel?	3630
:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: LODisruption:TIMestamp?	3631
:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: VTHReshold?	3632
:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: CDISruption?	3633
:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: LADisruption:CHANnel?	3634
:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: LADisruption:TIMestamp?	3635

1 **Introducing the FTB/IQS-81xx Transport Blazer SCPI Commands**

FTB/IQS-81xx Transport Blazer test modules are designed to specifically address R&D and manufacturing requirements, providing a new standard for SONET/SDH and Ethernet multiservice equipment testing.

FTB/IQS-81xx Transport Blazer modules allows to be supported and interchangeable on the FTB-400 Universal Test System, FTB-500 Universal Test System and IQS-600 Integrated Qualification System. This cross-platform support provides users with added flexibility by enabling them to select an appropriate platform that suits their testing needs.

This module can be remotely controlled using SCPI commands. You can also use these commands directly on the unit to build scripts to automate test processes.



IMPORTANT

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	0.00
<STRING RESPONSE DATA> <CHARACTER RESPONSE DATA>	No	"--"



IMPORTANT

Application currently does not support "O" as a suffix value. Hence, for all ODUO related commands, you should pass "100" as suffix (e.g. ODU100) and for ODUFlex commands you should pass "101" as suffix (e.g. ODU101).



IMPORTANT

Deleted Command:

SOURce:DATA:TELEcom:GFP:CONFig:UPI|FETHernet|FPPP|
TFCHAnnel|TFICON|TESCON|TGBEthernet|FMAPOS|TDVBASI|FRPR|
FFCBBW|TAFCHAnnel|FMPLSUCAST|FMPLSMCAST|FISIS|FIPV4|
FIPV6|FDVBASI|F64B66BETH

As this command's support through Simplified testSetup is been removed.

Key Features of FTB/IQS-8105 Transport Blazer

- Supports DS_n, PDH, SONET and SDH electrical rates up to 155 Mbit/s in a single-slot module
- Dual Rx DS_n
- Supports Test Summary command capabilities

Key Features of FTB/IQS-8115 Transport Blazer

- DS₀/E₀ to OC-48/STM-16 testing in a single module
- Supports SONET, SDH, DS_n, and PDH
- Supports Test Summary command capabilities

Key Features of FTB/IQS-8120/8130 Transport Blazer

- DS0/E0 to OC-192/STM-64 testing in a single module
- Supports SONET, SDH, DSn, PDH, and OTN testing
- Ethernet-over-SONET/SDH (EoS) testing via GFP, VCAT, and LCAS software options
- OTN forward error correction (FEC) and optical channel data unit (ODU) multiplex testing capabilities as per ITU-T G.709
- Supports Test Summary command capabilities

Key Features of FTB/IQS-8120NGE/8130NGE Transport Blazer

- DS0/E0 to OC-192/STM-64/OTU2; 10 Mbit/s to 10 Gbit/s LAN/WAN as well as 1x, 2x, 4x and 10x Fibre Channel testing in the industry's smallest form factor
- 10/100/1000 Mbit/s, GigE and 10 GigE (LAN\WAN) BERT
- ODU multiplexing (ODU1 into ODU2) testing
- Ethernet Dual Test Set configuration
- Round Trip Delay (RTD) measurement
- Service Disruption Time (SDT) measurement
- Frequency offset generation and analysis
- Fully integrated solution for assessing the performance of Ethernet transport networks, including RFC 2544 and BER test functionalities
- OTN forward error correction (FEC) and optical channel data unit (ODU)
- Signals for qualifying newly and efficiently mapped transport and datacom services over OTN

Introducing the FTB/IQS-81xx Transport Blazer SCPI Commands

Key Features of FTB/IQS-8140 Transport Blazer

- Supports circuit and packet ODUflex testing capabilities for OTN bandwidth optimization
- Ethernet-over-SONET/SDH (EoS) testing via optional support for GFP, VCAT and LCAS software options
- Supports complete EtherSAM™ (ITU-T Y.156sam) test suite
- Supports complete Carrier Ethernet services portfolio: PBB-TE, MPLS and IPv4/IPv6
- Far-End Alarm and Control feature
- Programmable Error/Alarm Injection
- Supports Test Summary and EoOTN command capabilities
- Supports ODU2 MUX and Block & Replace features
- Supports 10G->GFP->ODU2 features
- OTNRTD
- FC Overclocked
- Unframed BERT on all Optical Rates
- Dynamic Timeslot changing

Key Features of FTB/IQS-8140 Transport Blazer

- OC-768/STM-256 testing with STS-1/AU-3 granularity
- OTN testing at the OTU3 level including forward error correction (FEC) as per ITU-T G.709
- Compatible with any of EXFO's FTB-400 supported modules, including its industry-leading 40 Gbit/s fiber characterization modules
- Supports single and dual stage optical channel data unit
- Supports circuit and packet ODUflex testing capabilities for OTN bandwidth optimization

Introducing the FTB/IQS-81xx Transport Blazer SCPI Commands

Key Features of FTB/IQS-8140 Transport Blazer

- Service Disruption Time (SDT)
- Round Trip Delay (RTD)
- Programmable Error/Alarm Injection
- Supports Test Summary command capabilities
- Supports ODU3 MUX and Block & Replace features.
- Supports 10G->GFP->ODU2->ODU3 features.
- OTNRTD
- FC Overclocked.
- Dynamic Timeslot changing

Conventions

Before using the product described in this manual, 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 with FTB/IQS-81xx Transport Blazer SCPI Commands**

Selecting the Network Analyzer Mode

In the ToolBox main window, under Current Modules, press FTB-8130NGE once to select the module.

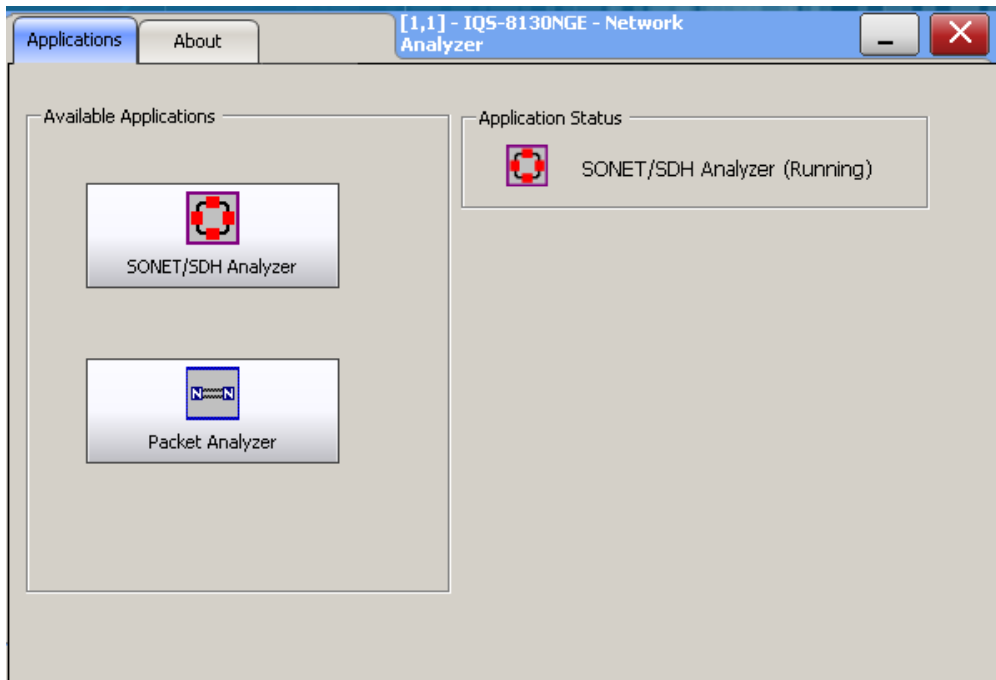
In the Online Applications bar, press the FTB-8130NGE Power Blazer application to start the **Network Analyzer**.

Perform the following steps to select SONET/SDH Analyzer or Packet Analyzer:

1. In the Available Applications, under the Applications tab, press **SONET/SDH Analyzer** to start the Smart User Interface (SUI) for Telecom test or press **Packet Analyzer** to start the Smart User Interface (SUI) for Datacom test.
2. Perform the same operation via SCPI using the following command:
`LINS[1..n]:INSTrument:SElect<wsp>ETHernet|SONetsdh`
3. Passing the parameter as ETHernet or SONetsdh, Packet Analyzer or SONET/SDH Analyzer application can be selected.

Getting Started with FTB/IQS-81xx Transport Blazer SCPI Commands

Selecting the Network Analyzer Mode



Note: *Telecom and Datacom applications cannot run simultaneously.*

Using SCPI Commands to create SONET OC-48 Test

This section contains a brief summary of the FTB/IQS-8130NGE/8130NGEv2 Transport Blazer specific commands.

Following steps will give an idea about the command and sequence of commands to execute and perform a specific test on FTB-400/FTB-500 and IQS-600 platforms.

1. *CLS

This command clears the register of module.

2. INST:CAT:FULL?

This command will detect the module attached to the FTB-400/FTB-500 or IQS-600 back panel and return the name of the module and its position with Unit number and Slot number.

For example “Transport Blazer (10.7 Gb/s) IQS-8130NGE”,10. This information we need to add with LINS keyword before each and every SCPI command. Following SCPI commands describes the use of LINS.

3. LINS10:INST:SElect SONetsdh

This command selects SONET/SDH Analyzer.

4. LINS10:INST:SElect?

This query returns SONETSDH as a selected instrument.

5. LINS10:SOURce:DATA:TELeom:CLEar

This command clears any existing running test.

6. LINS10:SOURce:DATA:TELeom:MODE NORMal

This command sets the test mode of the source as Normal.

7. LINS10:SOURce:DATA:TELeom:MODE?

This query returns the test mode of the source.

Getting Started with FTB/IQS-81xx Transport Blazer SCPI Commands

Using SCPI Commands to create SONET OC-48 Test

- 8.** LINS10:OUTPut:TELEcom:CONNector OPTical
This command selects the type of connector as Optical.
- 9.** LINS10:OUTPut:TELEcom:CONNector?
This query returns the type of connector.
- 10.** LINS10:SOURce:DATA:TELEcom:INTerface:TYPE OC48
This command selects the interface type as OC48.
- 11.** LINS10:SOURce:DATA:TELEcom:INTerface:TYPE?
This query returns the interface type.
- 12.** LINS10:SOURce:DATA:TELEcom:HOP:TYPE STS48C
This command selects the type of High Order Path (HOP) as STS48C.
- 13.** LINS10:SOURce:DATA:TELEcom:HOP:TYPE?
This query returns the type of High Order Path (HOP).
- 14.** LINS10:OUTPut:TELEcom:LASer ON
This command enables or disables the status of the laser.
- 15.** LINS10:OUTPut:TELEcom:LASer?
This query returns the current state of the laser.
- 16.** LINS10:SOURce:DATA:TELEcom:PATtern:TYPE PRBS2E9
This command selects the payload pattern type for the transmitter as PRBS2E9.
- 17.** LINS10:SOURce:DATA:TELEcom:PATtern:TYPE?
This query returns the payload pattern type of the transmitter.

Getting Started with FTB/IQS-81xx Transport Blazer SCPI Commands

Using SCPI Commands to create SONET OC-48 Test

18. LINS10:SOURce:DATA:TELEcom:TEST ON

This command starts the manual test.

19. LINS10:SOURce:DATA:TELEcom:TEST?

This query returns the status of the manual test.

20. LINS10:SOURce:DATA:TELEcom:SONet:ERRor:SECTion:MANual:TYPE
BERRor

This command sets the type of section error as B1.

21. LINS10:SOURce:DATA:TELEcom:SONet:ERRor:SECTion:MANual:TYPE?

This query returns the type of section error.

22. LINS10:SOURce:DATA:TELEcom:SONet:ERRor:SECTion:AMOUnt 15

This command sets the amount of section error to be injected into the instrument as 15.

23. LINS10:SOURce:DATA:TELEcom:SONet:ERRor:SECTion:AMOUnt?

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

24. LINS10:SOURce:DATA:TELEcom:SONet:ERRor:SECTion:INJect

This command injects the type of section error.

25. LINS10:SOURce:DATA:TELEcom:TEST OFF

This command stops the manual test.

For more information refer to **SONET/SDH SCPI Command Reference**.

This section contains information on commands description, syntax, examples, and reference to other commands.

3 **Introducing the SCPI over TCP/IP Service**

SCPI over TCP/IP Service is a Windows Service that continuously listens to a port (5024) at Server (IQS) on which modules to be tested are connected. For communication TCP/IP protocols are used. All Windows versions include Telnet Client and Telnet Server components. By using this, one can create a remote command console session on a remote computer. Commands can be executed just by logging on Server using Telnet interface.

Features

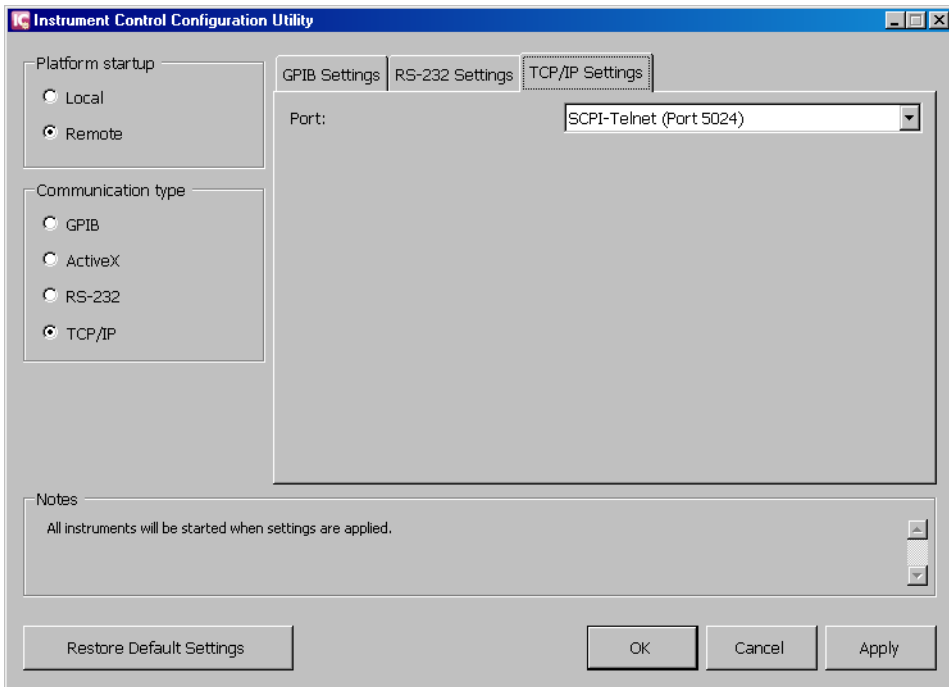
- Client from any platform (Windows/Linux/Unix) can connect to this service
- Client from any network domain can connect to the service
- Client can use Telnet components which are freely available on any platform
- One client/session can connect to one module at a time
- Client can execute single command or list of commands
- One user can connect to multiple modules through multiple sessions
- User can forcefully disconnect any already connected client

Introducing the SCPI over TCP/IP Service

Features

To activate SCPI over TCP/IP Service perform the following procedure:

1. Go to **Utilities** and click **Instrument Control Configuration**, the following screen will be displayed:

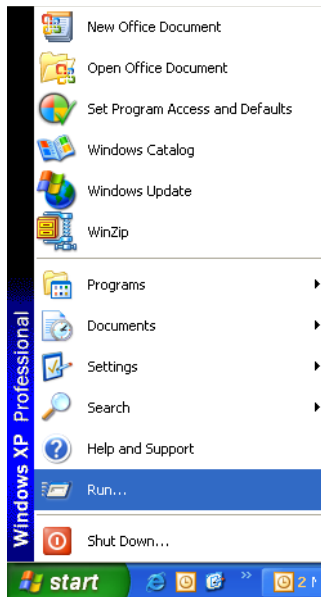


2. Make sure that the **TCP/IP** and **Remote** are selected.
3. Click on **Apply**, and then **OK**.

Steps to execute SCPI commands through SCPI over TCP/IP Service

Following is the procedure to connect SCPI over TCP/IP Service from any remote client through Windows Platform:

1. Click on Start, then Run.

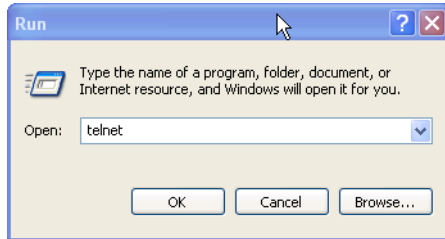


Introducing the SCPI over TCP/IP Service

Steps to execute SCPI commands through SCPI over TCP/IP Service

2. In **Open** window, type **telnet**.

It will open the telnet prompt as shown on second image below:



3. Connect to the Service using the following command:

open <IPADDRESS> <PORT>

Example: open 10.192.2.45 5024

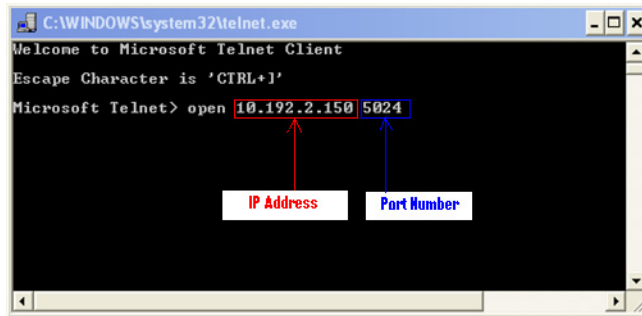
Red color denotes the IP address of IQS where SCPI over TCP/IP Service is running.

Blue color denotes the port number. #5024 is the default port number for SCPI over TCP/IP Service.

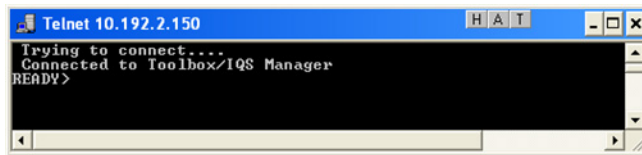
Introducing the SCPI over TCP/IP Service

Steps to execute SCPI commands through SCPI over TCP/IP Service

4. Press **Enter** to establish connection with the Service.



5. If connection is established successfully, the following window is displayed with READY> prompt.



Scenarios for release/block of module(s)

Scenario 1: Block

A module is blocked for other session once a connection is established with any session by executing either "CONNECT LINSxx" or a valid Instrument Command eg. "LINS10:SOURce:DATA:TELEcom:CLEAr".

Example: "LINS10:SOURce:DATA:TELEcom:CLEAr"

Once the above command is executed by client session "10.192.2.155:1364"; the module will be blocked for any other Session till the user does not release it by any of the following different activities:

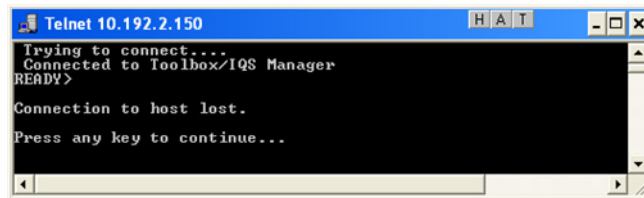
- Execute CLOSE LINSxx command to disconnect the link with the module.
- Executing CLOSE command to end current session.
- Closing the current session by clicking Close button on caption bar.
- Shutdown/Restart client computer.
- Network down.

Scenario 2: Release

- A module can be released forcefully by executing CLOSE LINSxx command from other session.

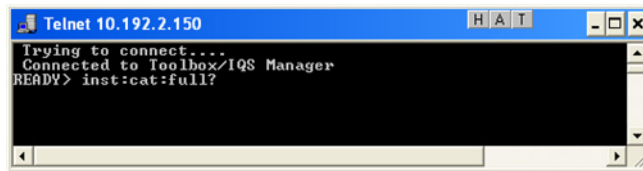
- Also one can kill a session by executing KILL LINSxx command (For more information please refer to KILL LINSxx).

If connection is not established, the following window is displayed.



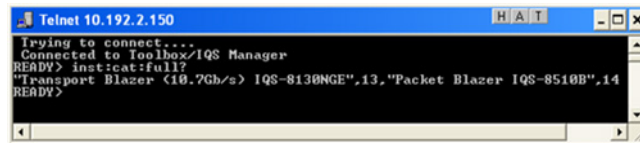
```
Telnet 10.192.2.150
Trying to connect...
Connected to Toolbox/IQS Manager
READY>
Connection to host lost.
Press any key to continue...
```

6. Once the connection is established, type or copy the SCPI command on Editor window to execute the command.



```
Telnet 10.192.2.150
Trying to connect...
Connected to Toolbox/IQS Manager
READY> inst:cat:full?
```

The response for the above command will appear as below:

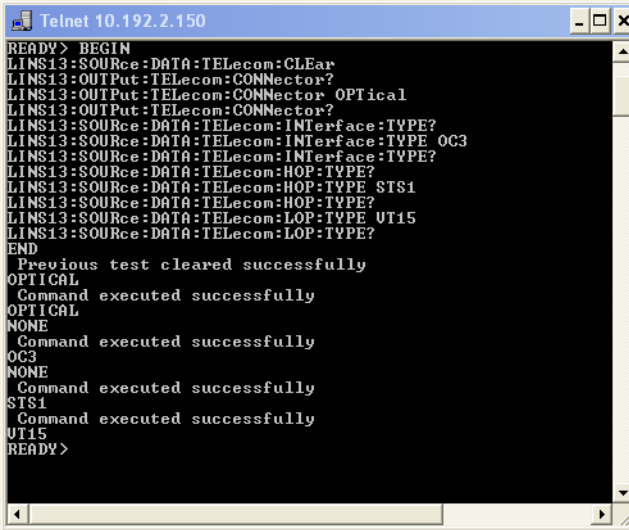


```
Telnet 10.192.2.150
Trying to connect...
Connected to Toolbox/IQS Manager
READY> inst:cat:full?
"Transport Blazer (10.7Gb/s) IQS-8130NGE",13,"Packet Blazer IQS-8510B",14
READY>
```

Introducing the SCPI over TCP/IP Service

Scenarios for release/block of module(s)

7. To execute multiple commands, copy the commands from any Script file and paste them between BEGIN ... END blocks on Editor window, and press **Enter**. Refer to Internal (Protocol) Commands of SCPI over TCP/IP Service on page 12 for more information.

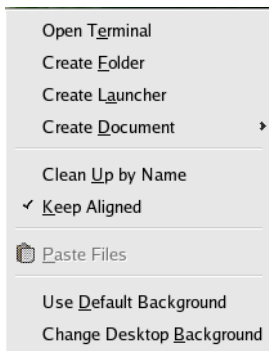


```
Telnet 10.192.2.150
READY> BEGIN
LINS13:SOURce:DATA:TELEcom:CLear
LINS13:OUTPut:TELEcom:CONNector?
LINS13:OUTPut:TELEcom:CONNector OPTical
LINS13:OUTPut:TELEcom:CONNector?
LINS13:SOURce:DATA:TELEcom:INTERface:TYPE?
LINS13:SOURce:DATA:TELEcom:INTERface:TYPE OC3
LINS13:SOURce:DATA:TELEcom:INTERface:TYPE?
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>
```

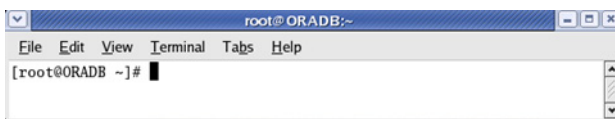
Note: *If a client tries to execute multiple commands without using BEGIN ...END blocks, it will still execute but with some ambiguity.*

Following is the procedure to connect SCPI over TCP/IP Service from any remote client through Linux Platform:

1. Right click on the Desktop, open the **Terminal**.



2. The command prompt will be displayed as below:



3. Type the following command

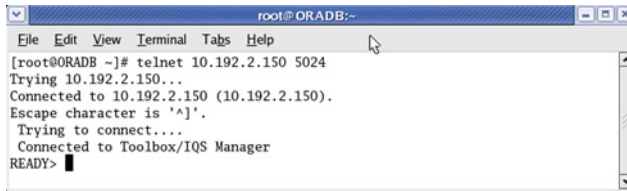
telnet<IPADDRESS> <PORT>

Example: telnet 10.192.3.27 5024

Introducing the SCPI over TCP/IP Service

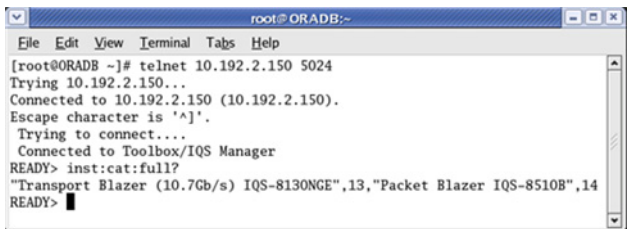
Scenarios for release/block of module(s)

4. The connection is established displaying the message “Connected to Toolbox/IQS Manager”.



```
root@ORADB:~  
File Edit View Terminal Tabs Help  
[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> █
```

5. Type a SCPI command on the Editor to execute.



```
root@ORADB:~  
File Edit View Terminal Tabs Help  
[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> inst:cat:full?  
"Transport Blazer (10.7Gb/s) IQS-8130NGE",13,"Packet Blazer IQS-8510B",14  
READY> █
```

6. Click on **X** to close the session.

Internal (Protocol) Commands of SCPI over TCP/IP Service

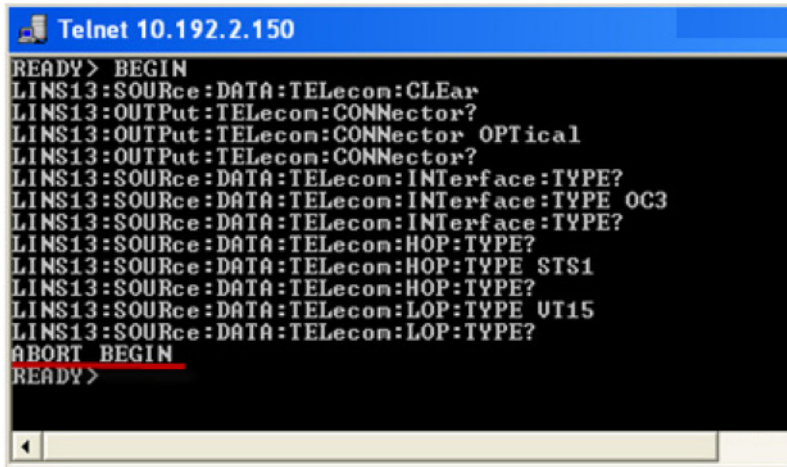
Note: *SCPI over TCP/IP Service Protocol commands are not case sensitive.*

Execution of single command

For executing single command, no protocol is required, simply type or paste the command on console and execute.

Execution of multiple commands (BEGIN and END block)

All commands should be entered within the “BEGIN” and “END” block.



```
Telnet 10.192.2.150
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: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 STS1
LINS13:SOURce:DATA:TELEcon:HOP:TYPE?
LINS13:SOURce:DATA:TELEcon:LOP:TYPE UT15
LINS13:SOURce:DATA:TELEcon:LOP:TYPE?
ABORT BEGIN
READY>
```

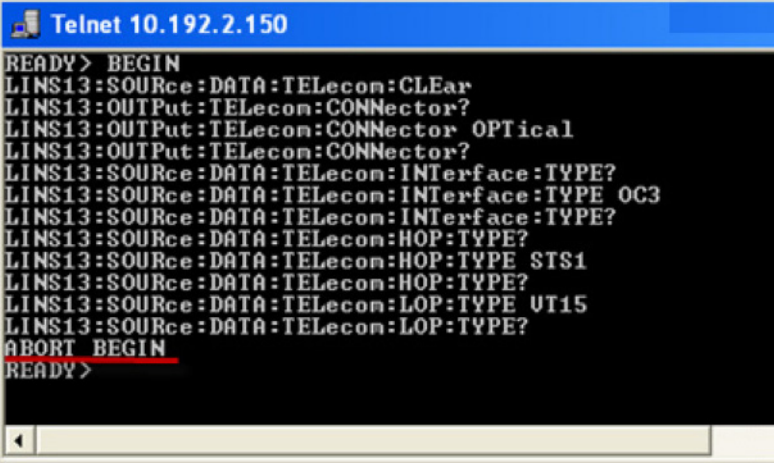
Note: *No Protocol commands are valid between Begin and End block except “ABORT BEGIN”*

Introducing the SCPI over TCP/IP Service

Internal (Protocol) Commands of SCPI over TCP/IP Service

ABORT BEGIN

ABORT BEGIN command will terminate the BEGIN ...END blocks, and returns to READY> prompt.



```
Telnet 10.192.2.150
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: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 STS1
LINS13:SOURce:DATA:TELEcon:HOP:TYPE?
LINS13:SOURce:DATA:TELEcon:LOP:TYPE UT15
LINS13:SOURce:DATA:TELEcon:LOP:TYPE?
ABORT BEGIN
READY>
```

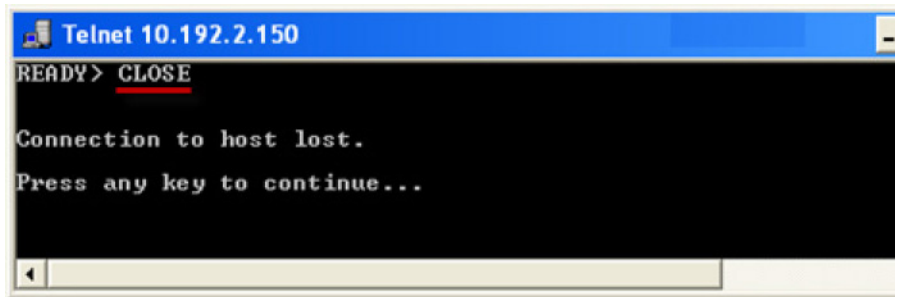
Note: *Except ABORT BEGIN any verb comes between BEGIN END block, that will be treated as a SCPI string.
Even if any commands script does not contain BEGIN END block, it will execute but with some ambiguity.*

Introducing the SCPI over TCP/IP Service

Internal (Protocol) Commands of SCPI over TCP/IP Service

CLOSE

Close command terminates the current client session of Telnet.

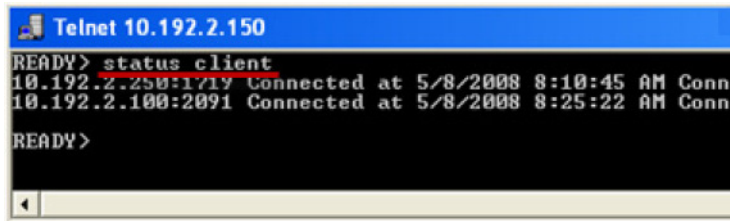


```
Telnet 10.192.2.150
READY> CLOSE

Connection to host lost.
Press any key to continue...
```

STATUS CLIENT

Status Client lists out the status of clients with connection time and modules connected



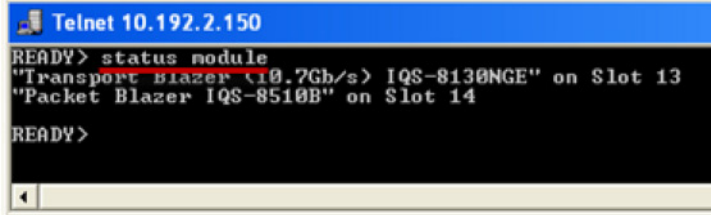
```
Telnet 10.192.2.150
READY> status client
10.192.2.250:1719 Connected at 5/8/2008 8:10:45 AM Conn
10.192.2.100:2091 Connected at 5/8/2008 8:25:22 AM Conn
READY>
```

Introducing the SCPI over TCP/IP Service

Internal (Protocol) Commands of SCPI over TCP/IP Service

STATUS MODULE

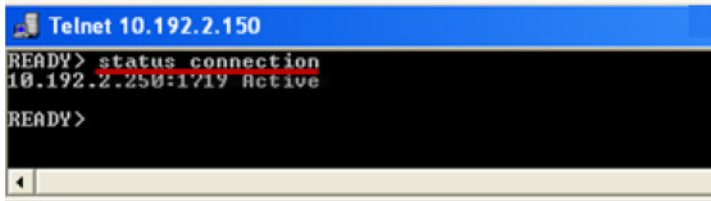
Status Module lists out the status of modules; with the slot numbers where they are connected.



```
Telnet 10.192.2.150
READY> status module
"Transport Blazer (10.7Gb/s) IQS-8130NGE" on Slot 13
"Packet Blazer IQS-8510B" on Slot 14
READY>
```

STATUS CONNECTION

Status Connection displays the connection along with status whether it is Active or Idle.



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

Note: *If any session is idle for a certain period (10 minutes), the service automatically changes its status to "Idle". The Idle time is configurable.*

CONNECT LINS

Syntax: CONNECT LINS[1..n]

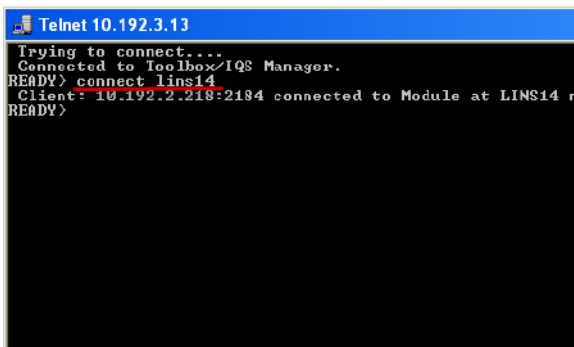
This command will allow the user to connect to different modules through TCP/IP. User can connect to multiple modules from a single Session. Where as, [1..n] denotes the unit number and slot number to which the session will connect.

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

- The module is already connected to different client session.
- Module is not present at said position(LINS[1..n])
- Not a valid LINS.

Note: *To ensure the backward compatibility,*

Connecting to a single module is not compulsorily requiring the “CONNECT LINS” command. A valid instrument command (eg. Lins10:SOURce:DATA:TELEcom:CLEar) for a valid lins position will still do fine for first module connectivity. It will compulsory only second module onwards.



```
Telnet 10.192.3.13
Trying to connect...
Connected to Toolbox/IQS Manager.
READY> connect_lins14
Client: 10.192.2.218:2184 connected to Module at LINS14 n
READY>
```

Introducing the SCPI over TCP/IP Service

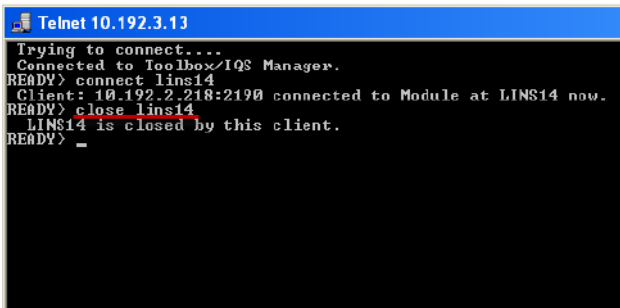
Internal (Protocol) Commands of SCPI over TCP/IP Service

CLOSE LINS

Syntax: CLOSE LINS[1..n]

This command will allow user to close an active connection. User can issue this command to close self as well as different client's connection with any Module.

The possible failure of this commands will be same as CONNECT LINS[1..n]



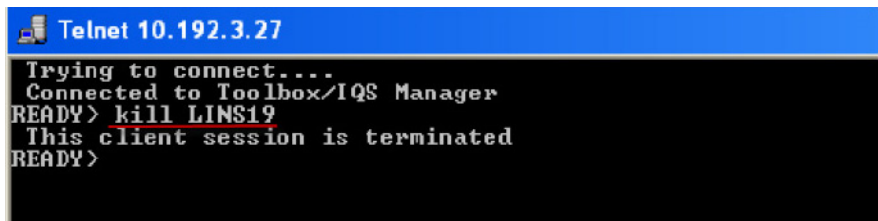
```
Telnet 10.192.3.13
Trying to connect...
Connected to Toolbox/IQS Manager.
READY> connect lins14
Client: 10.192.2.218:2190 connected to Module at LINS14 now.
READY> close lins14
LINS14 is closed by this client.
READY> _
```

KILL LINS

Syntax: Kill Lins[1..n]

This command allows any user to forcefully disconnect any existing client session.

Where as, [1..n] denotes the unit number and slot number to which the session is connected..



```
Telnet 10.192.3.27
Trying to connect...
Connected to Toolbox/IQS Manager
READY> kill LINS19
This client session is terminated
READY>
```

Note: *This command will terminate all active connections along with the Session; that it was connected before.*

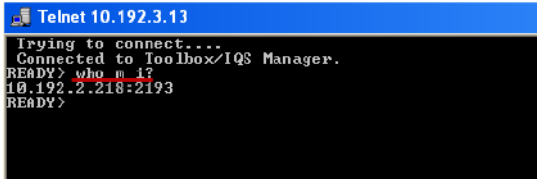
Introducing the SCPI over TCP/IP Service

Internal (Protocol) Commands of SCPI over TCP/IP Service

WHO M I?

Syntax – WHO M I?

It will retrieve current sessions IP along with Port, through which the session is communicating.



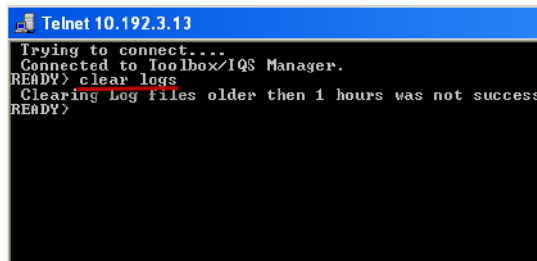
```
Telnet 10.192.3.13
Trying to connect...
Connected to ToolboX/IQS Manager.
READY> who m i?
10.192.2.218:2193
READY>
```

CLEAR LOGS

Syntax: CLEAR LOGS

It will clear all the log files generated under Client as well as Server sessions. Which are older than the 36 hours from the time of “CLEAR LOGS” command.

The Clear Log Duration is configurable



```
Telnet 10.192.3.13
Trying to connect...
Connected to ToolboX/IQS Manager.
READY> clear logs
Clearing Log files older then 1 hours was not success
READY>
```

3 SONET/SDH SCPI Command Reference

This chapter presents detailed information of the SCPI commands and queries supported by the FTB/IQS-81xx Transport Blazer.



IMPORTANT

Since the FTB-400, FTB-500 and IQS-600 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.

FTB-400 backplane identification number

|

1Y

|

Instrument slot number:

4-slot backplane: 0 to 3;

8-slot backplane: 0 to 7

For information on modifying unit identification, refer to the *FTB-400 Universal Test System User Guide*, *FTB-500 Universal Test System User Guide* and *IQS-600 Integrated Qualification System User Guide*.

Network Analyzer

:INSTrument[1..n]:SElect

Description This command selects the instrument. When a logical instrument is selected, all other logical instruments are unavailable for programming until selected.

At *RST, this value is set to None.

Syntax :INSTrument[1..n]:SElect<wsp>ETHernet|SONetsdh

Parameter(s) Instrument:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
ETHernet|SONetsdh.
Selects the instrument.
ETHernet, selects Ethernet analyzer as an instrument.
SONetsdh, selects SONET/SDH analyzer as an instrument.

:INSTrument[1..n]:SElect**Example(s)**

- * INST:SEL SON
- * INST:SEL? Returns SONETSDH

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * INSTrument[1..n]:SElect?
-

:INSTRument[1..n]:SElect?

Description	This query returns the selected instrument. At *RST, this value is set to None.
Syntax	:INSTRument[1..n]:SElect?
Parameter(s)	None
Response Syntax	<Instrument>
Response(s)	Instrument: The response data syntax for <Instrument> is defined as a <CHARACTER RESPONSE DATA> element. Selects an instrument. NONE, No instrument is selected. ETHERNET, Ethernet analyzer is selected as an instrument. SONETSDH, SONET/SDH analyzer is selected as an instrument.

:INSTrument[1..n]:SElect?**Example(s)**

- * INST:SEL SON
- * INST:SEL? Returns SONETSDH

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * INSTrument[1..n]:SElect
-

SONET/SDH Analyzer

Test Setup Command Reference

:SOURce[1..n]:DATA:TELEcom:MODE

Description	<p>This command sets the test mode for the source.</p> <p>At *RST, this value is set to NORMAl.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:MODE<wsp> NORMAl DRX</p>
Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NORMAl DRX.</p> <p>Sets the test mode for the instrument.</p> <p>NORMAl, selects the Normal mode.</p> <p>DRX, selects the Dual Rx mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:MODE NORMAl</p> <p>* SOUR:DATA:TEL:MODE? Returns NORMAL</p>
Note	<p>For 8120NGE/8130NGE/8130NGEv2 modules, choices are NORMAl DRX.</p> <p>For 8140 module, choice is NORMAl only.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:MODE?</p>

:SOURce[1..n]:DATA:TELEcom:MODE?

Description	This query returns the test mode for the source. At *RST, this value is set to NORMAl.
Syntax	:SOURce[1..n]:DATA:TELEcom:MODE?
Parameter(s)	None
Response Syntax	<Mode>
Response(s)	Mode: The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the test mode for the source. NORMAL, Normal mode is selected. DRX, Dual Rx mode is selected.
Example(s)	* SOUR:DATA:TEL:MODE NORMAl * SOUR:DATA:TEL:MODE? Returns NORMAL
See Also	* SOURce[1..n]:DATA:TELEcom:MODE

:OUTPut[1..n]:TELEcom:CONNector

Description	<p>This command selects the interface connector type for different data paths.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<p>:OUTPut[1..n]:TELEcom:CONNector<wsp> OPTical BNC BANTam RJ48C</p>
Parameter(s)	<p>Connector:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPTical BNC BANTam RJ48C.</p> <p>Selects the connector type.</p> <p>OPTical, selects an Optical as connector for OC-n/STM-n/OTN.</p> <p>BNC, selects the BNC as connector for STS-3e/STS-1e/STM-1e/STM-0e/DS3/E4/E3/E2/E1.</p> <p>BANTam, selects the Bantam as connector for DS1/E1.</p> <p>RJ48C, selects the RJ48 as connector for DS1/E1.</p>

:OUTPut[1..n]:TELEcom:CONNector

Example(s) * OUTP:TEL:CONN OPT
 * OUTP:TEL:CONN? Returns OPTICAL

Note FTB/IQS-8140 Transport Blazer does not support
 this command.

See Also * OUTPut[1..n]:TELEcom:CONNector?

:OUTPut[1..n]:TELEcom:CONNector?

Description This query returns an interface connector type for different data paths.

At *RST, the configuration is set to a device-dependent value.

Syntax :OUTPut[1..n]:TELEcom:CONNector?

Parameter(s) None

Response Syntax <Connector>

Response(s) Connector:
The response data syntax for <Connector> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the connector type.
OPTICAL, Optical connector is selected.
BNC, BNC connector is selected.
BANTAM, Bantam connector is selected.
RJ48C, RJ48 connector is selected.

:OUTPut[1..n]:TELEcom:CONNector?

Example(s) * OUTP:TEL:CONN OPT
 * OUTP:TEL:CONN? Returns OPTICAL

Note FTB/IQS-8140 Transport Blazer does not support
 this query.

See Also * OUTPut[1..n]:TELEcom:CONNector

:SOURce[1..n]:DATA:TELEcom:INTErface:TYPE

Description

This command selects the interface rate.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:INTErface:TYPE  
<wsp>OC3|OC12|OC48|OC192|OC768|STM1|  
STM4|STM16|STM64|STM256|STM0E|STM1E|  
STS3E|STS1E|OTU1|OTU1E|OTU2|OTU2E|DS3  
|DS1|E4RATE|E3RATE|E2RATE|E1RATE|E1BAN  
TAM|DS1RJ48|E1RJ48|OTU1e|OTU2e|OTU3|  
OTU1F|OTU2F|UFOC768STM256|  
UFOC192STM64|UFOTU3|UFOTU2|UFOTU2F|  
UFOTU2E|UFOTU1F|UFOTU1E
```

Parameter(s)

Interface:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

```
OC3|OC12|OC48|OC192|OC768|STM1|STM4|  
STM16|STM64|STM256|STM0E|STM1E|STS3E|  
STS1E|OTU1|OTU1E|OTU2|OTU2E|DS3|DS1|  
E4RATE|E3RATE|E2RATE|E1RATE|E1BANTAM|  
DS1RJ48|E1RJ48|OTU1e|OTU2e|OTU3|OTU1F|  
OTU2F|UFOC768STM256|UFOC192STM64|  
UFOTU3|UFOTU2|UFOTU2F|UFOTU2E|  
UFOTU1F|UFOTU1E.
```

Selects the interface rate.

OC3, selects the 155 Mbps transmission speed.

:SOURce[1..n]:DATA:TELecom:INTErface:TYPE

OC12, selects the 622 Mbps transmission speed.
OC48, selects the 2.5 Gbps transmission speed.
OC192, selects the 10 Gbps transmission speed.
OC768, selects the 40 Gbps transmission speed.
STM1, selects 155 Mbps transmission speed.
STM4, selects 622 Mbps transmission speed.
STM16, selects 2.5 Gbps transmission speed.
STM64, selects 10 Gbps transmission speed.
STM256, selects 40 Gbps transmission speed.
STM0E, selects 51.84 Mbps transmission speed.
STM1E, selects 155.52 Mbps transmission speed.
STS3E, selects the E4 (European standard for digital transmission-level 4) as interface rate.
STS1E, selects the E4 (European standard for digital transmission-level 4) as interface rate.
OTU1, selects the 2.66 Gbps transmission speed.
OTU1E, selects 11.04 Gbps transmission speed.
OTU2, selects the 10.71 Gbps transmission speed.
OTU2E, selects 11.09 Gbps transmission speed.
DS3, selects the DS3 (Digital Signal-level 3) as interface rate.
DS1, selects the DS1 (Digital Signal-level 1) as interface rate.
E4RATE, selects the E4 (European standard for digital transmission-level 4) as interface rate.
E3RATE, selects the E3 (European standard for digital transmission-level 3) as interface rate.
E2RATE, selects the E2 (European standard for digital transmission-level 2) as interface rate.
E1RATE, selects the E1 (European standard for digital transmission-level 1) as interface rate.

:SOURce[1..n]:DATA:TELEcom:INTERface:TYPE

E1BANTAM, selects the DS1 (Digital Signal-level 1) and E1 (European standard for digital transmission-level 1) as interface rate.

DS1RJ48, selects the DS1 (Digital Signal-level 1) as interface rate.

E1RJ48, selects the E1 (European standard for digital transmission-level 1) as interface rate.

OTU1e, selects DS1 (Digital Signal-level 1) as interface rate for RJ48C.

OTU2e, selects E1 (European standard for digital transmission-level 1) as interface rate for RJ48C.

OTU3, selects 43 Gbps transmission speed.

OTU1F, selects FC Overclocked OPTICAL rate of 11.270 089 285 714 3 Gb/s.

OTU2F, selects FC Overclocked for the data rate 11.317 642 405 063 3 Gb/s as interface rate for OPTICAL.

UFOC768STM256, selects the Unframed optical interface for the data 39.813G.

UFOC192STM64, selects the Unframed optical interface for the data 9.953G.

UFOTU3, selects the Unframed optical interface for the data 43.018G.

UFOTU2, selects the Unframed optical interface for the data 10.709G.

UFOTU2F, selects the Unframed optical interface for the data 11.317G.

UFOTU2E, selects the Unframed optical interface for the data 11.096G.

UFOTU1F, selects the Unframed optical interface for the data 11.270G.

UFOTU1E, selects the Unframed optical interface for the data 11.049G.

:SOURce[1..n]:DATA:TELEcom:INTErface:TYPE**Example(s)**

- * SOUR:DATA:TEL:INT:TYPE OC3
- * SOUR:DATA:TEL:INT:TYPE? Returns OC3

Note

8105-Supports-STM0E|STM1E|STS3E|STS1ED|S3|DS1|E4RATE|E3RATE|E2RATE|E1RATE
 8115-OC3|OC12|OC48|STM1|STM4|STM16|STM0E|STM1E|STS3E|STS1E|DS3|DS1|E4RATE|E3RATE|E2RATE|E1RATE|E1BANTAM|DS1RJ48|E1RJ48
 8120,8120NG,8120NGE-OC3|OC12|OC48|STM1|STM4|STM16|STM0E|STM1E|STS3E|STS1E|OTU1|DS3|DS1|E4RATE|E3RATE|E2RATE|E1RATE|E1BANTAM|DS1RJ48|E1RJ48
 8130NG,8130NGE-OC3|OC12|OC48|OC192|STM1|STM4|STM16|STM64|STM0E|STM1E|STS3E|STS1E|OTU1|OTU2|DS3|DS1|E4RATE|E3RATE|E2RATE|E1RATE|E1BANTAM|DS1RJ48|E1RJ48.
 8140-Supports-OTU3|UFOC768STM256|UFOTU3

See Also

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

:SOURce[1..n]:DATA:TELEcom:INTerface:TYPE?

Description	<p>This query returns the interface rate.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:INTerface:TYPE? <wsp></p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Interface></p>
Response(s)	<p>Interface: The response data syntax for <Interface> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the interface rate. OC3, 155 Mbps transmission speed is selected. OC12, 622 Mbps transmission speed is selected. OC48, 2.5 Gbps transmission speed is selected. OC192, 10 Gbps transmission speed is selected. STM1, 155 Mbps transmission speed is selected. STM4, 622 Mbps transmission speed is selected. STM16, 2.5 Gbps transmission speed is selected. STM64, 10 Gbps transmission speed is selected. STM0E, 51.84 Mbps transmission speed is selected.</p>

:SOURce[1..n]:DATA:TELEcom:INTERface:TYPE?

STM1E, 155.52 Mbps transmission speed is selected.

STS3E, Electrical Synchronous Transport Signal-Level 3 (STS3E) is selected as interface rate.

STS1E, Synchronous Transport Signal-Level 1 (STS1E) as interface rate as interface rate .

DS3, Digital Signal-level 3 (DS3) is selected as interface rate.

DS1, Digital Signal-level 1 (DS1) is selected as interface rate.

E4RATE, European standard for digital transmission-level 4 (E4) is selected as interface rate.

E3RATE, European standard for digital transmission-level 3 (E3) is selected as interface rate.

E2RATE, European standard for digital transmission-level 2 (E2) is selected as interface rate.

E1RATE, European standard for digital transmission-level 1 (E1) is selected as interface rate.

OTU1, 2.66 Gbps transmission speed is selected.

OTU1E, 11.04 Gbps transmission speed is selected.

OTU2, 10.71 Gbps transmission speed is selected.

OTU1E, 11.09 Gbps transmission speed is selected.

:SOURce[1..n]:DATA:TELEcom:INTERface:TYPE?

E1BANTAM, Digital Signal-level 1 (DS1) and European standard for digital transmission-level 1 (E1) as interface rate is selected.

DS1RJ48, Digital Signal-level (1DS1) as interface rate is selected.

E1RJ48, European standard for digital transmission-level 1 (E1) is selected as interface rate.

OTU1e, OTU1e (Optical Transport Unit 11.049 Gbps) is selected as interface rate.

OTU2e, OTU2e (Optical Transport Unit 11.096 Gbps) is selected as interface rate.

OTU3, OTU3 (Optical Transport Unit 43.018 Gbps) is selected as interface rate.

OTUIF, OTUIF (Optical Transport Unit 11.270 Gbps) is selected as interface rate.

OTU2F, OTU2F (Optical Transport Unit 11.317 Gbps) is selected as interface rate.

UFOC768STM256, UFOC768STM256 is selected as interface rate.

UFOC192STM64, UFOC192STM64 is selected as interface rate.

UFOTU3, UFOTU3 is selected as interface rate.

UFOTU2, UFOTU2 is selected as interface rate.

UFOTU2F, UFOTU2F is selected as interface rate.

UFOTU2E, UFOTU2E is selected as interface rate.

UFOTU1F, UFOTU1F is selected as interface rate.

UFOTU1E, UFOTU1E is selected as interface rate.

:SOURce[1..n]:DATA:TELEcom:INTerface:TYPE?**Example(s)**

- * SOUR:DATA:TEL:INT:TYPE OC3
- * SOUR:DATA:TEL:INT:TYPE? Returns OC3

Note(s)

8105-Supports-STM0E|STM1E|STS3E|STS1ED|S3|DS1|E4RATE|E3RATE|E2RATE|E1RATE
 8115-OC3|OC12|OC48|STM1|STM4|STM16|STM0E|STM1E|STS3E|STS1E|DS3|DS1|E4RATE|E3RATE|E2RATE|E1RATE|E1BANTAM|DS1RJ48|E1RJ48
 8120,8120NG,8120NGE-OC3|OC12|OC48|STM1|STM4|STM16|STM0E|STM1E|STS3E|STS1E|OTU1|DS3|DS1|E4RATE|E3RATE|E2RATE|E1RATE|E1BANTAM|DS1RJ48|E1RJ48
 8130NG,8130NGE-OC3|OC12|OC48|OC192|STM1|STM4|STM16|STM64|STM0E|STM1E|STS3E|STS1E|OTU1|OTU2|DS3|DS1|E4RATE|E3RATE|E2RATE|E1RATE|E1BANTAM|DS1RJ48|E1RJ48
 8140-Supports-OTU3|UFOC768STM256|UFOTU3

See Also

- * SOURce[1..n]:DATA:TELEcom:INTerface:TYPE

:SOURce[1..n]:DATA:TELEcom:HOP:TYPE

Description	<p>This command selects the High Order Path (HOP) type.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:HOP:TYPE <wsp>STS1 STS3C STS12C STS48C STS192C STS768C AU3 AU4 AU44C AU416C AU464C AU4256C</pre>
Parameter(s)	<p>Hop:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>STS1 STS3C STS12C STS48C STS192C STS768C AU3 AU4 AU44C AU416C AU464C AU4256C.</pre> <p>Selects the HOP (High Order Path) type. STS1, selects SPE (Synchronous Payload Envelope) type as STS-1.</p>

:SOURce[1..n]:DATA:TELecom:HOP:TYPE

STS3C, selects SPE type as STS-3C.
STS12C, selects SPE type as STS-12C.
STS48C, selects SPE type as STS-48C.
STS192C, selects SPE type as STS-192C.
STS768C, selects SPE type as STS-768C.
STS768C, selects the SPE type as STS768C.
AU3, selects AU-3 as AU (Administrative Unit) type.
AU4, selects AU-4 as AU type.
AU44C, selects AU-4-4C as AU type.
AU416C, selects AU-4-16C as AU type.
AU464C, selects AU-4-64C as AU type.
AU4256C, selects AU-4-256C as AU type.

:SOURce[1..n]:DATA:TELEcom:HOP:TYPE

Example(s)

- * SOUR:DATA:TEL:INT:TYPE OC3
- * SOUR:DATA:TEL:HOP:TYPE STS1
- * SOUR:DATA:TEL:HOP:TYPE? Returns STS1

Note

For **8120NGE/8130NGE/8130NGEv2** modules, choices are STS1 | STS3C | STS12C | STS48C | STS192C | AU3 | AU4 | AU44C | AU416C | AU464C.

For **8140 module**, choices are STS1 | STS3C | STS12C | STS48C | STS192C | STS768C | AU3 | AU4 | AU44C | AU416C | AU464C | AU4256C.

See Also

- * SOURce[1..n]:DATA:TELEcom:INTErface:TYPE
- * SOURce[1..n]:DATA:TELEcom:HOP:TYPE?

:SOURce[1..n]:DATA:TELEcom:HOP:TYPE?

Description	This query returns the High Order Path (HOP) type. At *RST, this value is device dependent.
Syntax	:SOURce[1..n]:DATA:TELEcom:HOP:TYPE?
Parameter(s)	None
Response Syntax	<Hop>
Response(s)	Hop: The response data syntax for <Hop> is defined as a <CHARACTER RESPONSE DATA> element. Returns the High Order Path (HOP) type. STS1, Synchronous Transport Signal-Level 1 (STS1) is selected as Synchronous Payload Envelope (SPE) type. STS3C, STS3C is selected as SPE type. STS12C, STS12C is selected as SPE type. STS48C, STS48C is selected as SPE type. STS192C, STS192C is selected as SPE type. STS768C, STS768C is selected as SPE type.

:SOURce[1..n]:DATA:TELEcom:HOP:TYPE?

AU3, AU-3 is selected as Administrative Unit (AU) type.

AU4, AU-4 is selected as AU type.

AU44C, AU-4-4C is selected as AU type.

AU416C, AU-4-16C is selected as AU type.

AU464C, AU-4-64C is selected as AU type.

AU4256C, AU4256C is selected as AU type.

Example(s)

- * SOUR:DATA:TEL:INT:TYPE OC3
- * SOUR:DATA:TEL:HOP:TYPE STS1
- * SOUR:DATA:TEL:HOP:TYPE? Returns STS1

See Also

- * SOURce[1..n]:DATA:TELEcom:INTerface:TYPE
 - * SOURce[1..n]:DATA:TELEcom:HOP:TYPE
-

:SOURce[1..n]:DATA:TELEcom:LOP:TYPE

Description This command selects the Low Order Path (LOP) type.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:LOP:TYPE
<wsp>VT15|VT2|VT6|TU11|TU12|TU2|TU3

Parameter(s) Lop:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
VT15|VT2|VT6|TU11|TU12|TU2|TU3.
Selects the LOP (Low Order Path) type.
VT15, selects VT1.5 (Virtual Tributary-1.5) as mapping type.
VT2, selects VT2 (Virtual Tributary-2) as mapping type.
VT6, selects VT6 (Virtual Tributary-6) as mapping type.
TU11, selects TU-11 (Tributary Unit) - 11 from TU group.
TU12, selects TU-12 from TU (Tributary Unit) group.
TU2, selects TU-2 from TU (Tributary Unit) group.
TU3, selects TU-3 from TU (Tributary Unit) group.

:SOURce[1..n]:DATA:TELEcom:LOP:TYPE

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:INT:TYPE OC3* SOUR:DATA:TEL:HOP:TYPE STS1* SOUR:DATA:TEL:LOP:TYPE VT15* SOUR:DATA:TEL:LOP:TYPE? Returns VT15
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:INTErface:TYPE* SOURce[1..n]:DATA:TELEcom:HOP:TYPE* SOURce[1..n]:DATA:TELEcom:LOP:TYPE?

:SOURCE[1..n]:DATA:TELEcom:LOP:TYPE?

Description	<p>This query returns the Low Order Path (LOP) type.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:LOP:TYPE?
Parameter(s)	None
Response Syntax	<Lop>
Response(s)	<p>Lop:</p> <p>The response data syntax for <Lop> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Low Order Path (LOP) type. NONE, No Low Order Path (LOP) type is selected.</p> <p>VT15, Virtual Tributary-1.5 (VT1.5) is selected as mapping type.</p> <p>VT2, Virtual Tributary-2 (VT2) is selected as mapping type.</p> <p>VT6, Virtual Tributary-6 (VT6) is selected as mapping type.</p> <p>TU11, Tributary Unit 11 (TU11) is selected as Tributary Unit (TU) type.</p> <p>TU12, Tributary Unit 12 (TU12) is selected as Tributary Unit (TU) type.</p>

:SOURce[1..n]:DATA:TELEcom:LOP:TYPE?

TU2, Tributary Unit 2 (TU2) is selected as Tributary Unit (TU) type.

TU3, Tributary Unit 3 (TU3) is selected as Tributary Unit (TU) type.

Example(s)

- * SOUR:DATA:TEL:INT:TYPE OC3
- * SOUR:DATA:TEL:HOP:TYPE STS1
- * SOUR:DATA:TEL:LOP:TYPE VT15
- * SOUR:DATA:TEL:LOP:TYPE? Returns VT15

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:INTERface:TYPE
 - * SOURce[1..n]:DATA:TELEcom:HOP:TYPE
 - * SOURce[1..n]:DATA:TELEcom:LOP:TYPE
-

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE**Description**

This command selects the payload for the Optical Data Unit (ODU).

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:ODU:TYPE
<wsp>O1OC48|O1ODU0OC12|O1ODU0STM4|
O1ODU0OC3|O1ODU0STM1|O1ODU0GFPETH|
O1STM16|O2O1ODU0|O2O1ODU0OC3|
O2ODU0|O2ODU0oc12|O2ODU0STM4|O2ODU0
OC3|O2ODU0STM1|O2ODU0GFPETH|O2O1OD
U0GFPETH|O2O1ODU0STM1|O2O1ODU0STM4|
O2O1ODU0OC12|O2OC192|O2STM64|O2O1OC
48|O2O1STM16|O2ODU1|O1ETHERNET|O2ETH
ERNET|O2GFP|O2GFPETH|O3ODU1|O3ODU1O
C48|O3ODU1STM16|O3ODU2|O3ODU2OC192|
O3ODU2STM64|O3O2ODU1|O3O2ODU1OC48|
O3O2ODU1STM16|O3OC768|O3O1ODU0OC12|
O3O1ODU0STM4|O3O1ODU0OC3|
O3O1ODU0STM1|O3O1ODU0GFPETH|O3ODU1
STM16|O3STM256|O3O1ODU0|O3O2O1ODU0|
O3O2O1ODU0STM4|O3O2O1ODU0OC12|
O3O2O1ODU0STM1|O3O2O1ODU0OC3|O3O2O
1ODU0GFPETH|O2OFLEXGFPETH|O3ODUFLEX
|O3OFLEXGFPETH|O2ODUFLEX|O3ODU2GFP|
O3ODU2GFPETH|O1ODU0MSDT|O2ODU0MSDT
|O3ODU0MSDT
```

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE

Parameter(s)

Odu:

The program data syntax for the parameter is defined as a <Character Program Data> element.

The allowed <Character Program Data> elements for this parameter are:

O1OC48|O1ODU0OC12|O1ODU0STM4|
O1ODU0OC3|O1ODU0STM1|O1ODU0GFPETH|
O1STM16|O2O1ODU0|O2O1ODU0OC3|
O2ODU0|O2ODU0oc12|O2ODU0STM4|O2ODU0
OC3|O2ODU0STM1|O2ODU0GFPETH|O2O1OD
U0GFPETH|O2O1ODU0STM1|O2O1ODU0STM4|
O2O1ODU0OC12|O2OC192|O2STM64|O2O1OC
48|O2O1STM16|O2ODU1|O1ETHERNET|O2ETH
ERNET|O2GFP|O2GFPETH|O3ODU1|O3ODU1O
C48|O3ODU1STM16|O3ODU2|O3ODU2OC192|
O3ODU2STM64|O3O2ODU1|O3O2ODU1OC48|
O3O2ODU1STM16|O3OC768|O3O1ODU0OC12|
O3O1ODU0STM4|O3O1ODU0OC3|
O3O1ODU0STM1|O3O1ODU0GFPETH|O3ODU1
STM16|O3STM256|O3O1ODU0|O3O2O1ODU0|
O3O2O1ODU0STM4|O3O2O1ODU0OC12|
O3O2O1ODU0STM1|O3O2O1ODU0OC3|O3O2O
1ODU0GFPETH|O2OFLEXGFPETH|O3ODUFLEX
|O3OFLEXGFPETH|O2ODUFLEX|O3ODU2GFP|
O3ODU2GFPETH|O1ODU0MSDT|O2ODU0MSDT
|O3ODU0MSDT.

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE

Selects the payload for ODU (Optical Channel Data) Unit.

O1OC48, Selects O1OC48 as the payload for ODU (Optical Channel Data) Unit.

O1ODU0OC12, Selects O1ODU0OC12 as the payload for ODU (Optical Channel Data) Unit.

O1ODU0STM4, Selects O1ODU0STM4 as the payload for ODU (Optical Channel Data) Unit.

O1ODU0OC3, Selects O1ODU0OC3 as the payload for ODU (Optical Channel Data) Unit.

O1ODU0STM1, Selects O1ODU0STM1 as the payload for ODU (Optical Channel Data) Unit.

O1ODU0GFPETH, Selects O1ODU0GFPETH as the payload for ODU (Optical Channel Data) Unit.

O1STM16, Selects O1STM16 as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0, Selects O2O1ODU0 as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0OC3, Selects O2O1ODU0OC3 as the payload for ODU (Optical Channel Data) Unit.

O2ODU0, Selects O2ODU0 as the payload for ODU (Optical Channel Data) Unit.

O2ODU0oc12, Selects O2ODU0oc12 as the payload for ODU (Optical Channel Data) Unit.

O2ODU0STM4, Selects O2ODU0STM4 as the payload for ODU (Optical Channel Data) Unit.

O2ODU0OC3, Selects O2ODU0OC3 as the payload for ODU (Optical Channel Data) Unit.

O2ODU0STM1, Selects O2ODU0STM1 as the payload for ODU (Optical Channel Data) Unit.

O2ODU0GFPETH, Selects O2ODU0GFPETH as the payload for ODU (Optical Channel Data) Unit.

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE

O2O1ODU0GFPETH, Selects O2O1ODU0GFPETH as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0STM1, Selects O2O1ODU0STM1 as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0STM4, Selects O2O1ODU0STM4 as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0OC12, Selects O2O1ODU0OC12 as the payload for ODU (Optical Channel Data) Unit.

O2OC192, Selects O2OC192 as the payload for ODU (Optical Channel Data) Unit.

O2STM64, Selects O2STM64 as the payload for ODU (Optical Channel Data) Unit.

O2O1OC48, Selects O2O1OC48 as the payload for ODU (Optical Channel Data) Unit.

O2O1STM16, Selects O2O1STM16 as the payload for ODU (Optical Channel Data) Unit.

O2ODU1, Selects O2ODU1 as the payload for ODU (Optical Channel Data) Unit.

O1ETHERNET, Selects O1ETHERNET as the payload for ODU (Optical Channel Data) Unit.

O2ETHERNET, Selects O2ETHERNET as the payload for ODU (Optical Channel Data) Unit.

O2GFP, Selects O2GFP as the payload for ODU (Optical Channel Data) Unit.

O2GFPETH, Selects O2GFPETH as the payload for ODU (Optical Channel Data) Unit.

O3ODU2GFP, Selects O3ODU2GFP as the payload for ODU (Optical Channel Data) Unit.

O3ODU2GFPETH, Selects O3ODU2GFPETH as the payload for ODU (Optical Channel Data) Unit.

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE

O3ODU1, Selects O3ODU1 as the payload for ODU (Optical Channel Data) Unit.

O3ODU1OC48, Selects O3ODU1OC48 as the payload for ODU (Optical Channel Data) Unit.

O3ODU1STM16, Selects O3ODU1STM16 as the payload for ODU (Optical Channel Data) Unit.

O3ODU2, Selects O3ODU2 as the payload for ODU (Optical Channel Data) Unit.

O3ODU2OC192, Selects O3ODU2OC192 as the payload for ODU (Optical Channel Data) Unit.

O3ODU2STM64, Selects O3ODU2STM64 as the payload for ODU (Optical Channel Data) Unit.

O3O2ODU1, Selects O3O2ODU1 as the payload for ODU (Optical Channel Data) Unit.

O3O2ODU1OC48, Selects O3O2ODU1OC48 as the payload for ODU (Optical Channel Data) Unit.

O3O2ODU1STM16, Selects O3O2ODU1STM16 as the payload for ODU (Optical Channel Data) Unit.

O3OC768, Selects O3OC768 as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0OC12, Selects O3O1ODU0OC12 as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0STM4, Selects O3O1ODU0STM4 as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0OC3, Selects O3O1ODU0OC3 as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0STM1, Selects O3O1ODU0STM as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0GFPETH, Selects O3O1ODU0GFPETH as the payload for ODU (Optical Channel Data) Unit.

O3ODU1STM16, Selects O3ODU1STM16 as the payload for ODU (Optical Channel Data) Unit.

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE

O3STM256, Selects O3STM256 as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0, Selects O3O1ODU0 as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0, Selects O3O2O1ODU0 as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0STM4, Selects O3O2O1ODU0STM4 as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0OC12, Selects O3O2O1ODU0OC12 as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0STM1, Selects O3O2O1ODU0STM1 as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0OC3, Selects O3O2O1ODU0OC3 as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0GFPETH, Selects O3O2O1ODU0GFPETH the payload for ODU (Optical Channel Data) Unit.

O2OFLEXGFPETH, Selects O2OFLEXGFPETH as the payload for ODU (Optical Channel Data) Unit.

O3ODUFLEX, Selects O3ODUFLEX as the payload for ODU (Optical Channel Data) Unit.

O3OFLEXGFPETH, Selects O3OFLEXGFPETH as the payload for ODU (Optical Channel Data) Unit.

O2ODUFLEX, Selects O2ODUFLEX as the payload for ODU (Optical Channel Data) Unit.

O1ODU0MSDT, Selects O1ODU0MSDT as the payload for ODU (Optical Channel Data) Unit.

O2ODU0MSDT, Selects O2ODU0MSDT as the payload for ODU (Optical Channel Data) Unit.

O3ODU0MSDT, Selects O3ODU0MSDT as the payload for ODU (Optical Channel Data) Unit.

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE**Example(s)**

SOUR:DATA:TEL:INT:TYPE OTU1
SOUR:DATA:TEL:ODU:TYPE O1OC48
SOUR:DATA:TEL:ODU:TYPE? Returns O1OC48

Note

O3GFP token is obsolete from now and new token O3ODU2GFP has been added.

O3GFPETH token is obsolete from now and new token O3ODU2GFPETH has been added.

See Also

SOURce[1..n]:DATA:TELEcom:INTErface:TYPE
SOURce[1..n]:DATA:TELEcom:ODU:TYPE?

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?

Description This query returns the payload for Optical Data Unit (ODU).

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:ODU:TYPE?

Parameter(s) None

Response Syntax <Odu>

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?**Response(s)**

Odu:

The response data syntax for <Odu> is defined as a <CHARACTER RESPONSE DATA> element. Returns the payload for the Optical Data Unit (ODU).

Returns the payload for Optical Channel Data Unit (ODU).

O1OC48, O1OC48 is selected as the payload for ODU (Optical Channel Data) Unit.

O1ODU0OC12, O1ODU0OC12 is selected as the payload for ODU (Optical Channel Data) Unit.

O1ODU0STM4, O1ODU0STM4 is selected as the payload for ODU (Optical Channel Data) Unit.

O1ODU0OC3, O1ODU0OC3 is selected as the payload for ODU (Optical Channel Data) Unit.

O1ODU0STM1, O1ODU0STM1 is selected as the payload for ODU (Optical Channel Data) Unit.

O1ODU0GFPETH, O1ODU0GFPETH is selected as the payload for ODU (Optical Channel Data) Unit.

O1STM16, O1STM16 is selected as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0, O2O1ODU0 is selected as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0OC3, O2O1ODU0OC3 is selected as the payload for ODU (Optical Channel Data) Unit.

O2ODU0, O2ODU0 is selected as the payload for ODU (Optical Channel Data) Unit.

O2ODU0oc12, O2ODU0oc12 is selected as the payload for ODU (Optical Channel Data) Unit.

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?

O2ODU0STM4, O2ODU0STM4 is selected as the payload for ODU (Optical Channel Data) Unit.

O2ODU0OC3, O2ODU0OC3 is selected as the payload for ODU (Optical Channel Data) Unit.

O2ODU0STM1, O2ODU0STM1 is selected as the payload for ODU (Optical Channel Data) Unit.

O2ODU0GFPETH, O2ODU0GFPETH is selected as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0GFPETH, O2O1ODU0GFPETH is selected as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0STM1, O2O1ODU0STM1 is selected as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0STM4, O2O1ODU0STM4 is selected as the payload for ODU (Optical Channel Data) Unit.

O2O1ODU0OC12, O2O1ODU0OC12 is selected as the payload for ODU (Optical Channel Data) Unit.

O2OC192, O2OC192 is selected as the payload for ODU (Optical Channel Data) Unit.

O2STM64, O2STM64 is selected as the payload for ODU (Optical Channel Data) Unit.

O2O1OC48, O2O1OC48 is selected as the payload for ODU (Optical Channel Data) Unit.

O2O1STM16, O2O1STM16 is selected as the payload for ODU (Optical Channel Data) Unit.

O2ODU1, O2ODU1 as the is selected payload for ODU (Optical Channel Data) Unit.

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?

O1ETHERNET, O1ETHERNET is selected as the payload for ODU (Optical Channel Data) Unit.

O2ETHERNET, O2ETHERNET is selected as the payload for ODU (Optical Channel Data) Unit.

O2GFP, O2GFP is selected as the payload for ODU (Optical Channel Data) Unit.

O2GFPETH, O2GFPETH is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODU2GFP, O3ODU2GFP is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODU2GFPETH, O3ODU2GFPETH is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODU1, O3ODU1 is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODU1OC48, O3ODU1OC48 is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODU1STM16, O3ODU1STM16 is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODU2, O3ODU2 is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODU2OC192, O3ODU2OC192 is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODU2STM64, O3ODU2STM64 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O2ODU1, O3O2ODU1 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O2ODU1OC48, O3O2ODU1OC48 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O2ODU1STM16, O3O2ODU1STM16 is selected as the payload for ODU (Optical Channel Data) Unit.

O3OC768, O3OC768 is selected as the payload for ODU (Optical Channel Data) Unit.

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?

O3O1ODU0OC12, O3O1ODU0OC12 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0STM4, O3O1ODU0STM4 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0OC3, O3O1ODU0OC3 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0STM1, O3O1ODU0STM is selected as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0GFPETH, O3O1ODU0GFPETH is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODU1STM16, O3ODU1STM16 is selected as the payload for ODU (Optical Channel Data) Unit.

O3STM256, O3STM256 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O1ODU0, O3O1ODU0 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0, O3O2O1ODU0 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0STM4, O3O2O1ODU0STM4 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0OC12, O3O2O1ODU0OC12 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0STM1, O3O2O1ODU0STM1 is selected as the payload for ODU (Optical Channel Data) Unit.

O3O2O1ODU0OC3, O3O2O1ODU0OC3 is selected as the payload for ODU (Optical Channel Data) Unit.

:SOURce[1..n]:DATA:TELEcom:ODU:TYPE?

O3O2O1ODU0GFPETH, O3O2O1ODU0GFPETH is selected as the payload for ODU (Optical Channel Data) Unit.

O2OFLEXGFPETH, O2OFLEXGFPETH is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODUFLEX, O3ODUFLEX is selected as the payload for ODU (Optical Channel Data) Unit.

O3OFLEXGFPETH, O3OFLEXGFPETH is selected as the payload for ODU (Optical Channel Data) Unit.

O2ODUFLEX, O2ODUFLEX is selected as the payload for ODU (Optical Channel Data) Unit.

O1ODU0MSDT, O1ODU0MSDT is selected as the payload for ODU (Optical Channel Data) Unit.

O2ODU0MSDT, O2ODU0MSDT is selected as the payload for ODU (Optical Channel Data) Unit.

O3ODU0MSDT, O3ODU0MSDT is selected as the payload for ODU (Optical Channel Data) Unit.

Example(s)

- * SOUR:DATA:TEL:INT:TYPE OTU3
- * SOUR:DATA:TEL:ODU:TYPE O3OC768
- * SOUR:DATA:TEL:ODU:TYPE? Returns O3OC768

See Also

- * SOURce[1..n]:DATA:TELEcom:INTErface:TYPE
 - * SOURce[1..n]:DATA:TELEcom:ODU:TYPE
-

:SOURce[1..n]:DATA:TELEcom:MAPPING:TYPE

Description	<p>This command selects the type of mapping.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:MAPPING:TYPE <wsp>DS3MAPPING DS1MAPPING E1MAPPING E2MAPPING E3MAPPING E4MAPPING</pre>
Parameter(s)	<p>Mapping:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DS3MAPPING DS1MAPPING E1MAPPING E2MAPPING E3MAPPING E4MAPPING.</p> <p>Selects the mapping type.</p> <p>DS3MAPPING, selects DS3 (Digital Signal-level 3) as mapping type.</p> <p>DS1MAPPING, selects DS1 as mapping type.</p> <p>E1MAPPING, selects E1 (European standard for digital transmission-level 1) as mapping type.</p> <p>E2MAPPING, selects E2 as mapping type.</p> <p>E3MAPPING, selects E3 as mapping type.</p> <p>E4MAPPING, selects E4 as mapping type.</p>

:SOURce[1..n]:DATA:TELEcom:MAPPING:TYPE

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:INT:TYPE OC3* SOUR:DATA:TEL:HOP:TYPE STS1* SOUR:DATA:TEL:MAPP:TYPE E3M* SOUR:DATA:TEL:MAPP:TYPE? Returns E3MAPPING
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:INTerface:TYPE* SOURce[1..n]:DATA:TELEcom:HOP:TYPE* SOURce[1..n]:DATA:TELEcom:MAPPING:TYPE?

:SOURce[1..n]:DATA:TELEcom:MAPPING:TYPE?

Description	This query returns the type of mapping. At *RST, this value is device dependent.
Syntax	:SOURce[1..n]:DATA:TELEcom:MAPPING:TYPE?
Parameter(s)	None
Response Syntax	<Mapping >
Response(s)	Mapping: The response data syntax for <Mapping> is defined as a <CHARACTER RESPONSE DATA> element. Returns the mapping type. NONE, No mapping type is selected. DS3MAPPING, Digital Signal-level 3 (DS3) is selected as mapping type. DS1MAPPING, DS1 is selected as mapping type. E1MAPPING, European standard for digital transmission-level 1 (E1) is selected as mapping type. E2MAPPING, E2 is selected as mapping type. E3MAPPING, E3 is selected as mapping type. E4MAPPING, E4 is selected as mapping type.

:SOURCE[1..n]:DATA:TELEcom:MAPPING:TYPE?**Example(s)**

- * SOUR:DATA:TEL:INT:TYPE OC3
- * SOUR:DATA:TEL:HOP:TYPE STS1
- * SOUR:DATA:TEL:MAPP:TYPE E3M
- * SOUR:DATA:TEL:MAPP:TYPE?

Returns E3MAPPING**Note**

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURCE[1..n]:DATA:TELEcom:INTERface:TYPE
- * SOURCE[1..n]:DATA:TELEcom:HOP:TYPE
- * SOURCE[1..n]:DATA:TELEcom:MAPPING:TYPE

:SOURce[1..n]:DATA:TELEcom:POSition

Description This command sets the mapping position.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:POSition<wsp>
LOPPosition|HOPPosition|DS1POSITION|
E3POSITION|E2POSITION|E1POSITION|
ODUPosition0|ODUPosition1|ODUPosition2|
ODUPFLEX,<Position1>,[<Position2>,
<Position3>,<Position4>]

:SOURCE[1..n]:DATA:TELEcom:POSITION**Parameter(s)**

Positionid:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> element for this parameter are

LOPPosition | HOPPosition | DS1POSITION |

E3POSITION | E2POSITION | E1POSITION |

ODUPosition0 | ODUPosition1 | ODUPosition2 |

ODUPFLEX

Selects the mapping position.

LOPPosition, selects LOP (Low Order Path) as mapping position.

HOPPosition, selects HOP (High Order Path) as mapping position.

DS1POSITION, selects DS1 (Digital Signal-level 1) as mapping position.

E3POSITION, selects E3 (European standard for digital transmission-level 3) as mapping position.

E2POSITION, selects E2 as mapping position.

E1POSITION, selects E1 as mapping position.

ODUPosition, selects ODU as mapping position.

ODUPosition1 selects ODU as mapping position.

ODUPosition2 selects the ODU as mapping position.

ODUPFLEX, selects ODUPFLEX as mapping position. (Only applicable for ODUFLEX) -

Tributary Slot position will be selected and unselected by same command. First execution of command with slot number will select tributary slot and again execution of same command will un-select tributary slot.

:SOURce[1..n]:DATA:TELEcom:POSITION

Position1:

The program data syntax for the second parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the first position value.

Position2:

The program data syntax for the third parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the second position value.

Position3:

The program data syntax for the fourth parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the third position value.

Position4:

The program data syntax for the fifth parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the fourth position value.

Example(s)

- * SOUR:DATA:TEL:HOP:TYPE STS1
- * SOUR:DATA:TEL:POS HOPP,1,3
- * SOUR:DATA:TEL:POS? HOPP Returns (1,3)

Note

The values of position varies with the selection of the position id.

Out of five parameter types, last three are optional.

:SOURCE[1..n]:DATA:TELEcom:POSITION?

Description	<p>This query returns the mapping position.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:POSITION<wsp> LOPPosition HOPPosition DS1POSITION E3POSITION E2POSITION E1POSITION ODUPosition0 ODUPosition1 ODUPosition2 ODUPFLEX</pre>
Parameter(s)	<p>Positionid:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter are LOPPosition HOPPosition DS1POSITION E3POSITION E2POSITION E1POSITION ODUPosition0 ODUPosition1 ODUPosition2 ODUPFLEX.</p> <p>Selects the mapping position.</p> <p>LOPPosition, selects LOP (Low Order Path) as mapping position.</p> <p>HOPPosition, selects HOP (High Order Path) as mapping position.</p> <p>DS1POSITION, selects DS1 (Digital Signal-level 1) as mapping position.</p>

:SOURce[1..n]:DATA:TELEcom:POSition?

E3POSITION, selects E3 (European standard for digital transmission-level 3) as mapping position.
E2POSITION, selects E2 as mapping position.
E1POSITION, selects E1 as mapping position.
ODUPosition0, selects ODU0 as mapping position.
ODUPosition1, selects ODU1 as mapping position.
ODUPosition2, selects ODU2 as mapping position.
ODUPFLEX, selects ODUPFLEX as mapping position.

Response Syntax <Position>

Response(s) Position:
The response data syntax for <Position> is defined as a <ARBITRARY ASCII RESPONSE DATA> element.

Example(s)

- * SOUR:DATA:TEL:HOP:TYPE STS1
- * SOUR:DATA:TEL:POS HOPP,1,3
- * SOUR:DATA:TEL:POS? HOPP Returns (1,3)

Note The values of position varies with the selection of the position id.

See Also * SOURce[1..n]:DATA:TELEcom:POSition

:SOURCE[1..n]:CLOCK:SOURCE

Description	<p>This command sets the clock mode for the source.</p> <p>At *RST, this value is set to INTERNAL.</p>
Syntax	<code>:SOURCE[1..n]:CLOCK:SOURCE<wsp>INTERNAL EXTERNAL RECOVERED BPLANE</code>
Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: INTERNAL EXTERNAL RECOVERED BPLANE.</p> <p>Sets the clock mode for the source.</p> <p>INTERNAL, selects the Internal 32 MHz clock.</p> <p>EXTERNAL, selects the External clock.</p> <p>RECOVERED, selects the Recovered clock.</p> <p>BPLANE, selects the Backplane clock.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:CLOC:SOUR INT* SOUR:CLOC:SOUR? Returns INTERNAL
See Also	<ul style="list-style-type: none">* SOURCE[1..n]:CLOCK:SOURCE?

:SOURce[1..n]:CLOCK:SOURce?

Description This query returns the clock mode for the source.

At *RST, this value is set to INTERNAL.

Syntax :SOURce[1..n]:CLOCK:SOURce?

Parameter(s) None

Response Syntax <Mode>

Response(s) Mode:
The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the clock mode.
INTERNAL, Internal 32MHz clock is selected.
EXTERNAL, External clock is selected.
RECOVERED, Recovered clock is selected.
BPLANE, Backplane clock is selected.

Example(s) * SOUR:CLOC:SOUR INT
* SOUR:CLOC:SOUR? Returns INTERNAL

See Also * SOURce[1..n]:CLOCK:SOURce

:SOURce[1..n]:DATA:TELEcom:ELECtrical:CODE**Description**

This command selects the line code for the transmitter.

At *RST, this value is device dependent.

Syntax

:SOURce[1..n]:DATA:TELEcom:ELECtrical:CODE
<wsp>AMI|B8ZS|B3ZS1|HDB3|CMI

Parameter(s)

Code:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

AMI|B8ZS|B3ZS1|HDB3|CMI.

Selects the line code.

AMI, selects the AMI (Alternate Mark Inversion) code external timing line code.

B8ZS, selects the B8ZS (Bipolar with 8 Zero Substitution) code external timing line code.

B3ZS1, selects the B3ZS (Bipolar with 3 Zero Substitution) external timing line code.

HDB3, selects the HDB3 (High Density Bipolar 3 Code) external timing line code.

CMI, selects the CMI (Coded Mark Inversion) external timing line code.

:SOURce[1..n]:DATA:TELecom:ELECtrical:CODE

Example(s) * SOUR:DATA:TEL:ELEC:CODE AMI
 * SOUR:DATA:TEL:ELEC:CODE? Returns AMI

Note FTB/IQS-8140 Transport Blazer does not support
 this command.

See Also * SOURce[1..n]:DATA:TELecom:ELECtrical:
 CODE?

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
CODE?**

Description	<p>This query returns the line code for the transmitter.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:ELECtrical: CODE?
Parameter(s)	None
Response Syntax	<Code>
Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as a <CHARACTER RESPONSE DATA> element. Returns the line code.</p> <p>AMI, Alternate Mark Inversion (AMI) code external timing line code is selected.</p> <p>B8ZS, Bipolar with 8 Zero Substitution (B8ZS) code external timing line code is selected.</p> <p>B3ZS1, Bipolar with 3 Zero Substitution (B3ZS) external timing line code is selected.</p> <p>HDB3, High Density Bipolar 3 Code (HDB3) external timing line code is selected.</p> <p>CMI, Coded Mark Inversion (CMI) external timing line code is selected.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
CODE?**

Example(s) * SOUR:DATA:TEL:ELEC:CODE AMI
 * SOUR:DATA:TEL:ELEC:CODE? Returns AMI

Note FTB/IQS-8140 Transport Blazer does not support
 this query.

See Also * SOURce[1..n]:DATA:TELEcom:ELECtrical:
 CODE

:SENSe[1..n]:DATA:TELEcom:ELECtrical:CODE

Description This command sets the line code for the receiver.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:ELECtrical:CODE
<wsp>AMI|B8ZS|B3ZS1|HDB3|CMI

Parameter(s) Code:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
AMI|B8ZS|B3ZS1|HDB3|CMI.
Sets the line code.
AMI, selects the AMI (Alternate Mark Inversion) code external timing line code.
B8ZS, selects the B8ZS (Bipolar with 8 Zero Substitution) code external timing line code.
B3ZS1, Bipolar with 3 Zero Substitution (B3ZS) external timing line code is selected.
HDB3, selects the HDB3 (High Density Bipolar 3 Code) external timing line code.
CMI, selects the CMI (Coded Mark Inversion) external timing line code.

SONET/SDH SCPI Command Reference

SONET/SDH Analyzer

:SENSe[1..n]:DATA:TELecom:ELECtrical:CODE

Example(s)	* SENS:DATA:TEL:ELEC:CODE AMI * SENS:DATA:TEL:ELEC:CODE? Returns AMI
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SENSe[1..n]:DATA:TELecom:ELECtrical:CODE?

:SENSe[1..n]:DATA:TELEcom:ELECtrical:CODE?

Description	This query returns the line code for the receiver. At *RST, this value is device dependent.
Syntax	:SENSe[1..n]:DATA:TELEcom:ELECtrical:CODE?
Parameter(s)	None
Response Syntax	<Code>
Response(s)	Code: The response data syntax for <Code> is defined as a <CHARACTER RESPONSE DATA> element. Returns the line code. AMI, Alternate Mark Inversion (AMI) code external timing line code is selected. B8ZS, Bipolar with 8 Zero Substitution (B8ZS) code external timing line code is selected. B3ZS1, Bipolar with 3 Zero Substitution (B3ZS) external timing line code is selected. HDB3, High Density Bipolar 3 Code (HDB3) external timing line code is selected. CMI, Coded Mark Inversion (CMI) external timing line code is selected.

:SENSe[1..n]:DATA:TELEcom:ELECtrical:CODE?

Example(s)	* SENS:DATA:TEL:ELEC:CODE AMI * SENS:DATA:TEL:ELEC:CODE? Returns AMI
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SENSe[1..n]:DATA:TELEcom:ELECtrical:CODE

:SOURce[1..n]:DATA:TELEcom:ELECtrical:LBO

Description	<p>This command selects Line Build Out (LBO) value to meet the interface requirements over the full range of cable lengths.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ELECtrical:LBO <wsp>R0TO225 R225TO450 CSIMulation P30 P24 P18 P12 P06 P00 N75 N150 N225</pre>
Parameter(s)	<p>Lbo:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: R0TO225 R225TO450 CSIMulation P30 P24 P18 P12 P06 P00 N75 N150 N225.</p> <p>Selects the LBO (Line Build Out) value.</p> <p>R0TO225, selects the LBO (Line Build Out) value as 0 to 225 feet range.</p> <p>R225TO450, selects the LBO value as 225 to 450 feet range.</p>

:SOURce[1..n]:DATA:TELEcom:ELECtrical:LBO

CSimulation, selects the LBO value as cable simulation 900 ft.

P30, selects the LBO value as +3.0 dBdsx (533-655 ft).

P24, selects the LBO value as +2.4 dBdsx (399-533 ft).

P18, selects the LBO value as +1.8 dBdsx (266-399 ft).

P12, selects the LBO value as +1.2 dBdsx (133-266 ft).

P06, selects the LBO value as +0.6 dBdsx (0-133 ft).

P00, selects the LBO value as 0.0 dBdsx.

N75, selects the LBO value as -7.5 dBdsx.

N150, selects the LBO value as -15.0 dBdsx.

N225, selects the LBO value as -22.5 dBdsx.

Example(s)

* SOUR:DATA:TEL:ELEC:LBO R0TO225

* SOUR:DATA:TEL:ELEC:LBO? Returns R0TO225

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:ELECtrical:LBO?

:SOURce[1..n]:DATA:TELEcom:ELECtrical:LBO?

Description	This query returns the Line Build Out (LBO) value. At *RST, this value is device dependent.
Syntax	:SOURce[1..n]:DATA:TELEcom:ELECtrical:LBO?
Parameter(s)	None
Response Syntax	<Lbo>
Response(s)	Lbo: The response data syntax for <Lbo> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Line Build Out (LBO) value. R0TO225, 0 to 225 feet range is selected. R225TO450, 225 to 450 feet range is selected. CSIMulation, Cable Simulation 900 ft is selected. P30, +3.0 dBdsx (533-655 ft) is selected. P24, +2.4 dBdsx (399-533 ft) is selected. P18, +1.8 dBdsx (266-399 ft) is selected. P12, +1.2 dBdsx (133-266 ft) is selected. P06, +0.6 dBdsx (0-133 ft) is selected. P00, 0.0 dBdsx is selected. N75, -7.5 dBdsx is selected.

:SOURce[1..n]:DATA:TELEcom:ELECtrical:LBO?

N150, -15.0 dBdsx is selected.

N225, -22.5 dBdsx is selected.

Example(s)

* SOUR:DATA:TEL:ELEC:LBO R0TO225

* SOUR:DATA:TEL:ELEC:LBO? Returns R0TO225

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:ELECtrical:LBO

**:SENSe[1..n]:DATA:TELEcom:ELECtrical:
TERMination****Description**

This command selects the termination mode.

At *RST, this value is device dependent.

Syntax

:SENSe[1..n]:DATA:TELEcom:ELECtrical:
TERMination<wsp>TERM|MON|BRIDge

Parameter(s)

Termination:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

TERM|MON|BRIDge.

Selects the termination mode.

TERM, selects the Term mode.

MON, selects the Mon mode.

BRIDge, selects the Bridge mode.

:SENSe[1..n]:DATA:TELEcom:ELECtrical: TERMination

Example(s)	* SENS:DATA:TEL:ELEC:TERM TERM * SENS:DATA:TEL:ELEC:TERM? Returns TERM
Note	This command does not support FTB/IQS-8140 Transport Blazer.
See Also	* SENSe[1..n]:DATA:TELEcom:ELECtrical: TERMination?

**:SENSe[1..n]:DATA:TELEcom:ELECtrical:
TERMination?**

Description	This query returns the termination mode. At *RST, this value is device dependent.
Syntax	:SENSe[1..n]:DATA:TELEcom:ELECtrical: TERMination?
Parameter(s)	None
Response Syntax	<Termination>
Response(s)	Termination: The response data syntax for <Termination> is defined as a <CHARACTER RESPONSE DATA> element. Returns the termination mode. TERM, Term mode is selected. MON, Mon mode is selected. BRIDGE, Bridge mode is selected.

**:SENSe[1..n]:DATA:TELEcom:ELECtrical:
TERMination?**

Example(s) * SENS:DATA:TEL:ELEC:TERM TERM
 * SENS:DATA:TEL:ELEC:TERM? Returns TERM

Note This query does not support FTB/IQS-8140
 Transport Blazer.

See Also * SENSe[1..n]:DATA:TELEcom:ELECtrical:
 TERMination?

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

This command selects the line code.

At *RST, this value is device dependent.

Syntax

:OUTPut[1..n]:TELEcom:CODE<wsp>AMI|
B8ZS|HDB3

Parameter(s)

Lcode:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

AMI|B8ZS|HDB3.

Selects the line code.

AMI, selects the AMI (Alternate Mark Inversion) code external timing line code.

B8ZS, selects the B8ZS (Bipolar with 8 Zero Substitution) code external timing line code.

HDB3, selects the HDB3 (High Density Bipolar 3 Code) external timing line code.

Example(s)

* OUTP:TEL:CODE AMI

* OUTP:TEL:CODE? Returns AMI

See Also

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

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

Description	This query returns the line code. At *RST, this value is device dependent.
Syntax	:OUTPut[1..n]:TELEcom:CODE?
Parameter(s)	None
Response Syntax	<Code>
Response(s)	Code: The response data syntax for <Code> is defined as a <CHARACTER RESPONSE DATA> element. Returns the line code. AMI, Alternate Mark Inversion (AMI) code external timing line code is selected. B8ZS, Bipolar with 8 Zero Substitution (B8ZS) code external timing line code is selected. HDB3, High Density Bipolar 3 Code (HDB3) external timing line code is selected.
Example(s)	* OUTP:TEL:CODE AMI * OUTP:TEL:CODE? Returns AMI
See Also	* OUTPut[1..n]:TELEcom:CODE

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

Description This command selects the framing of the signal.

At *RST, this value is device dependent.

Syntax :OUTPut[1..n]:TELEcom:FRAMing<wsp>SF|
ESF1|PCM30|PCM30C4|PCM31|PCM31C4

Parameter(s) Framing:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
SF|ESF1|PCM30|PCM30C4|PCM31|PCM31C4.
Selects the frame coding of the signal.
SF, selects the SF (Superframe) as frame coding of the signal.
ESF1, selects the ESF (Extended Superframe) as frame coding of the signal.
PCM30, selects the PCM30 (Pulse Code Modulation) as frame coding of the signal.
PCM30C4, selects the PCM30 (Pulse Code Modulation) CRC (Cyclic Redundancy Check) as frame coding of the signal.

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

PCM31 selects the PCM31 (Pulse Code Modulation) as frame coding of the signal.
PCM31C4, selects the PCM31(Pulse Code Modulation) CRC (Cyclic Redundancy Check) as frame coding of the signal.

Example(s)

- * OUTP:TEL:FRAM SF
- * OUTP:TEL:FRAM? Returns SF

See Also

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

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

Description	This query returns the framing of the signal. At *RST, this value is device dependent.
Syntax	:OUTPut[1..n]:TELEcom:FRAMing?
Parameter(s)	None
Response Syntax	<Framing>
Response(s)	Framing: The response data syntax for <Framing> is defined as a <CHARACTER RESPONSE DATA> element. Returns the available frame coding of the signal. SF, Superframe (SF) is selected. ESF1, Extended Superframe (ESF) is selected. PCM30, Pulse Code Modulation (PCM30) is selected. PCM30C4, Pulse Code Modulation (PCM30) Cyclic Redundancy Check (CRC) is selected. PCM31, Pulse Code Modulation (PCM31) is selected.

:OUTPut[1..n]:TELecom:FRAMing?

PCM31C4, Pulse Code Modulation (PCM31)
Cyclic Redundancy Check (CRC) is selected.

Example(s)

- * OUTP:TEL:FRAM SF
- * OUTP:TEL:FRAM? Returns SF

See Also

- * OUTPut[1..n]:TELecom:FRAMing
-

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

This command selects the external timing interface output level for the output port.

At *RST, this value is set to NONE.

Syntax

:OUTPut[1..n]:TELEcom:LEVel<wsp>NONE|
DS1LEVEL|E1BNC|S2MHZ

Parameter(s)

Level:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

NONE|DS1LEVEL|E1BNC|S2MHZ.

Selects the external timing interface output level.
NONE, No external timing interface output level is selected.

DS1LEVEL, selects DS1 (Digital Signal-level 1) as external timing interface output level.

E1BNC, selects BNC (Bayonet-Neill-Concelman) as external timing interface output level.

:OUTPut[1..n]:TELecom:LEVel

S2MHZ, selects 2 MHZ as external timing interface output level.

Example(s)

- * OUTP:TEL:LEV DS1L
- * OUTP:TEL:LEV? Returns DS1LEVEL

See Also

- * OUTPut[1..n]:TELecom:LEVel?
-

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

Description	<p>This query returns the external timing interface output level for the output port.</p> <p>At *RST, this value is set to NONE.</p>
Syntax	:OUTPut[1..n]:TELEcom:LEVel?
Parameter(s)	None
Response Syntax	<Level>
Response(s)	<p>Level:</p> <p>The response data syntax for <Level> is defined as a <CHARACTER RESPONSE DATA> element. Returns the external timing interface output level.</p> <p>NONE, No external timing interface output level is selected.</p> <p>DS1LEVEL, Digital Signal-level 1 (DS1) as external timing interface output level is selected.</p> <p>E1BNC, Bayonet-Neill-Concelman (BNC) as external timing interface output level is selected.</p>

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

S2MHZ, 2 MHZ as external timing interface output level is selected.

Example(s)

- * OUTP:TEL:LEV DS1L
- * OUTP:TEL:LEV? Returns DS1LEVEL

See Also

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

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

Description	<p>This command selects the termination mode.</p> <p>At *RST, this value is set to TERM.</p>
Syntax	<pre>:OUTPut[1..n]:TELEcom:TERMination<wsp> TERM MON BRIDge</pre>
Parameter(s)	<p>Termination:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>TERM MON BRIDge.</p> <p>Selects the termination mode.</p> <p>TERM, selects the Term as termination mode.</p> <p>MON, selects the Mon as termination mode.</p> <p>BRIDge, selects the Bridge as termination mode.</p>
Example(s)	<ul style="list-style-type: none">* OUTP:TEL:TERM TERM* OUTP:TEL:TERM? Returns TERM
See Also	<ul style="list-style-type: none">* OUTPut[1..n]:TELEcom:TERMination?

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

Description	This query returns the termination mode. At *RST, this value is set to TERM.
Syntax	:OUTPut[1..n]:TELEcom:TERMination?
Parameter(s)	None
Response Syntax	<Termination>
Response(s)	Termination: The response data syntax for <Termination> is defined as a <CHARACTER RESPONSE DATA> element. Returns the termination mode. TERM, Term as termination mode is selected. MON, Mon as termination mode is selected. BRIDGE, Bridge as termination mode is selected.
Example(s)	* OUTP:TEL:TERM TERM * OUTP:TEL:TERM? Returns TERM
See Also	* OUTPut[1..n]:TELEcom:TERMination

:SOURCE[1..n]:DATA:TELEcom:TEST

Description	<p>This command starts or stops the manual test.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:TEST <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Starts or stops the manual test.</p>
Example(s)	<pre>* SOUR:DATA:TEL:TEST ON * SOUR:DATA:TEL:TEST? Returns 1</pre>
See Also	<pre>* SOURCE[1..n]:DATA:TELEcom:TEST?</pre>

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

Description	This query returns the status of the manual test. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:TEST?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of manual test.
Example(s)	* SOUR:DATA:TEL:TEST ON * SOUR:DATA:TEL:TEST? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:TEST

:SOURce[1..n]:DATA:TELEcom:RESet

Description	This command resets the test. This command is an event and has no associated *RST condition or query form.
Syntax	:SOURce[1..n]:DATA:TELEcom:RESet
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:RES
See Also	* SOURce[1..n]:DATA:TELEcom:HRESet

:SOURce[1..n]:DATA:TELEcom:HRESet

Description	<p>This command resets the history status of the test.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:HRESet
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:HRES
See Also	* SOURce[1..n]:DATA:TELEcom:RESet

:SOURce[1..n]:DATA:TELEcom:MOUNT

Description	This command mounts the test. This command is an event and has no associated *RST condition or query form.
Syntax	:SOURce[1..n]:DATA:TELEcom:MOUNT
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:MOUN
See Also	* SOURce[1..n]:DATA:TELEcom:CLEar

:SOURce[1..n]:DATA:TELEcom:CLEAr

Description	This command clears the current running test. This command is an event and has no associated *RST condition or query form.
Syntax	:SOURce[1..n]:DATA:TELEcom:CLEAr
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:CLE
See Also	* SOURce[1..n]:DATA:TELEcom:MOUNT

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

Description	<p>This command loads the previously saved configuration setting as per given path.</p> <p>This command is an event and is not associated with *RST condition or query form.</p>
Syntax	<pre>:CONFig[1..n]:DATA:TELEcom:LOAD <wsp> <Path></pre>
Parameter(s)	<p>Path:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the file name.</p>
Example(s)	<pre>* CONF:DATA:TEL:LOAD "D:\Program Files\ EXFO\ToolBox\UserFiles\SonetSDHAnalyzerG2\ Configuration\test.cfg"</pre>
See Also	<pre>* CONFig[1..n]:DATA:TELEcom:SAVE</pre>

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

Description	<p>This command saves the current test configuration at a given path.</p> <p>This command is an event and is not associated with *RST condition or query form.</p>
Syntax	<pre>:CONFig[1..n]:DATA:TELEcom:SAVE<wsp> <Path></pre>
Parameter(s)	<p>Path:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the file name.</p>
Example(s)	<pre>* CONF:DATA:TEL:SAVE "D:\ToolBox\User Files\SonetSDHAnalyzerG2\Configuration\ test.cfg"</pre>
See Also	<pre>* CONFig[1..n]:DATA:TELEcom:LOAD</pre>

**:SOURce[1..n]:DATA:TELEcom:BACKground:
OTN**

Description	<p>This command selects the Optical Data Unit (ODU) multiplexed background traffic.</p> <p>At *RST, this value is set to PRBS31.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:BACKground: OTN <wsp>AIS NCLIENT1 PRBS31</pre>
Parameter(s)	<p>Background:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS NCLIENT1 PRBS31.</p> <p>Selects the background traffic.</p> <p>AIS, selects the AIS (Alarm Indication Signal) as background traffic.</p> <p>NCLIENT1, selects the Null Client as background traffic.</p> <p>PRBS31, selects the PRBS31 (Pseudo Random Bit Sequence) Pattern as background traffic.</p>

:SOURce[1..n]:DATA:TELecom:BACKground: OTN

Example(s) * SOUR:DATA:TEL:BACK:OTN PRBS31
 * SOUR:DATA:TEL:BACK:OTN? Returns PRBS31

Note * The above parameter is valid, only through
 ODU1 path.
 * FTB/IQS-8140 Transport Blazer does not
 support this command.

See Also * SOURce:DATA:TELecom:BACKground:OTN?

**:SOURce[1..n]:DATA:TELEcom:BACKground:
OTN?**

Description	<p>This query returns the Optical Data Unit (ODU) multiplexed background traffic.</p> <p>At *RST, this value is set to PRBS31.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:BACKground: OTN?
Parameter(s)	None
Response Syntax	<Background>
Response(s)	<p>Background:</p> <p>The response data syntax for <Background> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the background traffic.</p> <p>AIS, Alarm Indication Signal (AIS) is selected as background traffic.</p> <p>NCLIENT1, Null client (NCLIENT) as background traffic is selected.</p> <p>PRBS31, Pseudo Random Bit Sequence (PRBS31) Pattern as background traffic is selected.</p>

:SOURce[1..n]:DATA:TELEcom:BACKground: OTN?

Example(s)	* SOUR:DATA:TEL:BACK:OTN PRBS31 * SOUR:DATA:TEL:BACK:OTN? Returns PRBS31
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce:DATA:TELEcom:BACKground:OTN

**:SOURce[1..n]:DATA:TELEcom:BACKground:
OTNFlex****Description**

This command allows the selection of the ODU FLEX multiplexed background traffic.

At *RST, this value set to AIS.

Syntax

:SOURce[1..n]:DATA:TELEcom:BACKground:
OTNFlex<wsp>AIS|NCLIENT|PRBS31
|UNALlocated

**:SOURce[1..n]:DATA:TELEcom:BACKground:
OTNFlex**

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS NCLIENT PRBS31 UNALlocated.</p> <p>Allows the selection of the ODU FLEX multiplexed background traffic.</p> <p>AIS, selects AIS as the ODU FLEX multiplexed background traffic.</p> <p>NCLIENT,selects NULL Client (All Zeros) as the ODU FLEX multiplexed background traffic.</p> <p>PRBS31,selects PRBS31 pattern as the ODU FLEX multiplexed background traffic.</p> <p>UNALlocated,selects Unallocated as the ODU FLEX multiplexed background traffic.</p>
Example(s)	<p>SOURce:DATA:TELEcom:BACKground:OTNFlex AIS</p>
See Also	<p>SOURce[1..n]:DATA:TELEcom:BACKground:OTN</p>

**:SOURce[1..n]:DATA:TELEcom:BACKground:
OTNFlex?**

Description	This query returns the selected ODU FLEX multiplexed background traffic. At *RST, this value set to AIS.
Syntax	:SOURce[1..n]:DATA:TELEcom:BACKground: OTNFlex?
Parameter(s)	None
Response Syntax	<Type>

:SOURce[1..n]:DATA:TELEcom:BACKground: OTNFlex?

Response(s)

Type:

The response data syntax for <Type> is defined as a <CHARACTER RESPONSE DATA> element. Returns the selected ODU FLEX multiplexed background traffic.

AIS, AIS is selected as the ODU FLEX multiplexed background traffic.

NCLIENT, NULL Client (All Zeros) is selected as the ODU FLEX multiplexed background traffic.

PRBS31, PRBS31 pattern is selected as the ODU FLEX multiplexed background traffic.

UNALlocated, Unallocated is selected as the ODU FLEX multiplexed background traffic.

Example(s)

SOURce:DATA:TELEcom:BACKground:OTNFlex
AIS

SOURce:DATA:TELEcom:BACKground:OTNFlex?
Returns AIS

See Also

SOURce[1..n]:DATA:TELEcom:BACKground:
OTNFlex?

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

Description	<p>This command selects the High Order Path (HOP) background traffic.</p> <p>At *RST, this value is set to EQUIPPED1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:BACKground: SDHSonet:HOP <wsp>AIS UNEQUIPPED1 EQUIPPED1</pre>
Parameter(s)	<p>Background:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS UNEQUIPPED1 EQUIPPED1.</p> <p>Selects the High Order Path (HOP) background traffic.</p> <p>AIS, selects the AIS (Alarm Indication Signal) as background traffic.</p> <p>UNEQUIPPED1, selects the Unequipped as background traffic.</p>

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

EQUIPPED1, selects the Equipped as background traffic.

Example(s)

* SOUR:DATA:TEL:BACK:SDHS:HOP EQU1
* SOUR:DATA:TEL:BACK:SDHS:HOP?
Returns EQUIPPED1

See Also

* SOURce:DATA:TELEcom:BACKground:
SDHSonet:HOP?

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

Description	<p>This query returns the High Order Path (HOP) background traffic.</p> <p>At *RST, this value is set to EQUIPPED1.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:BACKground: SDHSonet:HOP?
Parameter(s)	None
Response Syntax	<Background>
Response(s)	<p>Background:</p> <p>The response data syntax for <Background> is defined as a <CHARACTER RESPONSE DATA> 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>

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

EQUIPPED1, Equipped as background traffic is selected.

Example(s)

* SOUR:DATA:TEL:BACK:SDHS:HOP EQU1
* SOUR:DATA:TEL:BACK:SDHS:HOP?
Returns EQUIPPED1

See Also

* SOURce:DATA:TELEcom:BACKground:
SDHSonet:HOP

**:SOURce[1..n]:DATA:TELEcom:BACKground:
SDHSonet:LOP**

Description	<p>This command selects the Low Order Path (LOP) background traffic.</p> <p>At *RST, this value is set to EQUIPPED1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:BACKground: SDHSonet:LOP <wsp>AIS UNEQUIPPED1 EQUIPPED1</pre>
Parameter(s)	<p>Background:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS UNEQUIPPED1 EQUIPPED1.</p> <p>Selects the Low Order Path (LOP) background traffic.</p> <p>AIS, selects the AIS (Alarm Indication Signal) as background traffic.</p> <p>UNEQUIPPED1, selects the Unequipped as background traffic.</p> <p>EQUIPPED1, selects the Equipped as background traffic.</p>

:SOURce[1..n]:DATA:TELEcom:BACKground: SDHSonet:LOP

Example(s)	* SOUR:DATA:TEL:BACK:SDHS:LOP EQU1 * SOUR:DATA:TEL:BACK:SDHS:LOP? Returns EQUIPPED1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce:DATA:TELEcom:BACKground: SDHSonet:LOP?

**:SOURce[1..n]:DATA:TELEcom:BACKground:
SDHSonet:LOP?**

Description	<p>This query returns the Low Order Path (LOP) background traffic.</p> <p>At *RST, this value is set to EQUIPPED1.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:BACKground: SDHSonet:LOP?
Parameter(s)	None
Response Syntax	<Background>
Response(s)	<p>Background:</p> <p>The response data syntax for <Background> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the Low Order Path (LOP) 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>

:SOURce[1..n]:DATA:TELEcom:BACKground: SDHSonet:LOP?

Example(s)	* SOUR:DATA:TEL:BACK:SDHS:LOP EQU1 * SOUR:DATA:TEL:BACK:SDHS:LOP? Returns EQUIPPED1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce:DATA:TELEcom:BACKground: SDHSonet:LOP

**:SOURce[1..n]:DATA:TELEcom:BACKground:
PDH**

Description	<p>This command selects the payload background traffic.</p> <p>At *RST, this value is set to AIS.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:BACKground: PDH<wsp>AIS ZERO</pre>
Parameter(s)	<p>Background:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS ZERO.</p> <p>Selects the payload background traffic.</p> <p>AIS, selects the AIS (Alarm Indication Signal) as background traffic.</p> <p>ZERO, selects ZERO as background traffic.</p>
Example(s)	<pre>* SOUR:DATA:TEL:BACK:PDH ZERO * SOUR:DATA:TEL:BACK:PDH? Returns ZERO</pre>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this command.</p>
See Also	<pre>* SOURce:DATA:TELEcom:BACKground:PDH?</pre>

**:SOURce[1..n]:DATA:TELEcom:BACKground:
PDH?**

Description	This query returns the payload background traffic. At *RST, this value is set to AIS.
Syntax	:SOURce[1..n]:DATA:TELEcom:BACKground: PDH?
Parameter(s)	None
Response Syntax	<Background>
Response(s)	Background: The response data syntax for <Background> is defined as a <CHARACTER RESPONSE DATA> element. Returns the payload background traffic. AIS, Alarm Indication Signal (AIS) as background traffic is selected. ZERO, ZERO as background traffic is selected.

**:SOURce[1..n]:DATA:TELecom:BACKground:
PDH?****Example(s)**

- * SOUR:DATA:TEL:BACK:PDH ZERO
- * SOUR:DATA:TEL:BACK:PDH? Returns ZERO

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce:DATA:TELecom:BACKground:PDH
-

:SOURce[1..n]:DATA:TELEcom:BACKground: BULK

Description	<p>This command enables or disables the Bulk Filled Overwrite Enable feature.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:BACKground: BULK<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Bulk Filled Overwrite Enable feature.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:BACK:BULK ON* SOUR:DATA:TEL:BACK:BULK? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:BACKground: BULK?

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

Description	<p>This query returns the status of Bulk Filled Overwrite Enable feature.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:BACKground: BULK?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the Bulk Filled Overwrite Enable feature.</p>
Example(s)	<p>* SOUR:DATA:TEL:BACK:BULK ON</p> <p>* SOUR:DATA:TEL:BACK:BULK? Returns 1</p>
See Also	* SOURce[1..n]:DATA:TELEcom:BACKground: BULK

**:SOURce[1..n]:DATA:TELEcom:BACKground:
COMPUtation**

Description	<p>This command selects the computation method.</p> <p>At *RST, this value is set to M0M1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:BACKground: COMPUtation<wsp>M1ONLY M0M1</p>
Parameter(s)	<p>Background:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: M1ONLY M0M1.</p> <p>Selects the payload computation method.</p> <p>M1ONLY, selects the M1 as computation method.</p> <p>M0M1, selects the M1 and M0 as computation method.</p>
Example(s)	<p>* SOUR:DATA:TEL:BACK:COMP M0M1</p> <p>* SOUR:DATA:TEL:BACK:COMP? Returns M0M1</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this command.</p>
See Also	<p>* SOURce:DATA:TELEcom:BACKground: COMPUtation?</p>

**:SOURce[1..n]:DATA:TELEcom:BACKground:
COMPUtation?**

Description	This query returns the computation method. At *RST, this value is set to M0M1.
Syntax	:SOURce[1..n]:DATA:TELEcom:BACKground: COMPUtation?
Parameter(s)	None
Response Syntax	<Background>
Response(s)	Background: The response data syntax for <Background> is defined as a <CHARACTER RESPONSE DATA> element. Returns the computation method. M1ONLY, M1 as computation method is selected. M0M1, M1 and M0 as computation method is selected.

**:SOURce[1..n]:DATA:TELEcom:BACKground:
COMPUtation?**

Example(s)	* SOUR:DATA:TEL:BACK:COMP M0M1 * SOUR:DATA:TEL:BACK:COMP? Returns M0M1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce:DATA:TELEcom:BACKground: COMPUtation

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

Description	<p>This query returns the time elapsed since the beginning of the test.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:TEST:TIME?
Parameter(s)	None
Response Syntax	<Time>
Response(s)	<p>Time:</p> <p>The response data syntax for <Time> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the time elapsed since the beginning of the test.</p>
Example(s)	<p>* SOUR:DATA:TEL:TEST ON</p> <p>* FETC:DATA:TEL:TEST:TIME?</p> <p>Returns the time elapsed since the beginning of the test.</p>
See Also	* SOURce[1..n]:DATA:TELEcom:TEST

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

Description	<p>This query returns the history status of any alarms/errors related to the tests such as Physical (Port), OTN, SONET/SDH, DS_n/PDH, Next Generation, Pattern, and Other.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:TEST:GLOBal: HISTory?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><History></p>
Response(s)	<p>History: The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element. Returns the history status of any alarms/errors related to the test. PRESENT, indicates that at least one alarm/error has occurred. PRESENT, indicates that at least one alarm/error has occurred. ABSENT, indicates that no alarm/error occurred. INACTIVE, indicates that the test did not run yet.</p>

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

ABSENT, indicates that no alarm/error occurred.
INACTIVE, indicates that the test did not run yet.

Example(s)

* FETC:DATA:TEL:TEST:GLOB:HIST?

Returns the history status of any alarms/errors related to the test.

See Also

* FETCh:DATA:TELEcom:TEST:GLOBal:
CURRent?

:FETCh[1..n]:DATA:TELEcom:TEST:GLOBal: CURRent?

Description	<p>This query returns the current status of any alarms/errors related to the tests such as Physical (Port), OTN, SONET/SDH, DSn/PDH, Next Generation, Pattern, and Other.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:TEST:GLOBal: CURRent?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Current></p>
Response(s)	<p>Current: The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element. Returns the current status of any alarms/errors related to the test. PRESENT, indicates that at least one alarm/error has occurred in the last second.</p>

**:FETCh[1..n]:DATA:TELecom:TEST:GLOBal:
CURRent?**

ABSENT, indicates that there is no alarm/error.
INACTIVE, indicates that the test is not running.

Example(s)

* FETC:DATA:TEL:TEST:GLOB:CURR?
Returns the current status of any alarms/errors related to the test.

See Also

* FETCh:DATA:TELecom:TEST:GLOBal:HISTory?

:SENSe[1..n]:DATA:TELeom:TEST:START:TIME

Description	<p>This command sets a specific time the created test case will start automatically.</p> <p>At *RST, this value is set to "1970-01-01 05:30:00".</p>
Syntax	<p>:SENSe[1..n]:DATA:TELeom:TEST:START:TIME <wsp><Time></p>
Parameter(s)	<p>Time: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets a specific time the created test case will start automatically.</p>
Example(s)	<p>* SENS:DATA:TEL:TEST:START:TIME "2007-09-10 05:30:00" * SENS:DATA:TEL:TEST:START:TIME? Returns "2007-09-10 05:30:00"</p>
See Also	<p>* SENSe[1..n]:DATA:TELeom:TEST:START:TIME? * SENSe[1..n]:DATA:TELeom:TEST:START:TIME:ENABled</p>

:SENSe[1..n]:DATA:TELEcom:TEST:START:TIME?

Description	This query returns the specific time the created test case will start automatically. At *RST, this value is set to "1970-01-01 05:30:00"
Syntax	:SENSe[1..n]:DATA:TELEcom:TEST:START:TIME?
Parameter(s)	None
Response Syntax	<Time>
Response(s)	Time: The response data syntax for the parameter is defined as a <STRING RESPONSE DATA> element. Returns the specific time the created test case will start automatically.
Example(s)	* SENS:DATA:TEL:TEST:START:TIME "2007-09-10 05:30:00" * SENS:DATA:TEL:TEST:START:TIME? Returns "2007-09-10 05:30:00"
See Also	* SENSe[1..n]:DATA:TELEcom:TEST:START:TIME * SENSe[1..n]:DATA:TELEcom:TEST:START:TIME:ENABled

:SENSe[1..n]:DATA:TELEcom:TEST:START:TIME:ENABLEd

Description	<p>This command enables or disables the Start Time configuration.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:TEST:START:TIME:ENABLEd<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Start Time configuration.</p>
Example(s)	<pre>* SENS:DATA:TEL:TEST:START:TIME "2007-09-10 05:30:00" * SENS:DATA:TEL:TEST:START:ENAB ON * SENS:DATA:TEL:TEST:START:TIME:ENAB? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:TEST:START:TIME * SENSe[1..n]:DATA:TELEcom:TEST:START:TIME:ENABLEd?</pre>

**:SENSe[1..n]:DATA:TELEcom:TEST:START:
TIME:ENABled?**

Description	<p>This query returns the status of the Start Time configuration.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:TEST:START:TIME: ENABled?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the Start Time configuration.</p>
Example(s)	<pre>* SENS:DATA:TEL:TEST:START:TIME "2007-09-10 05:30:00" * SENS:DATA:TEL:TEST:START:TIME:ENAB ON * SENS:DATA:TEL:TEST:START:TIME:ENAB? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:TEST:START:TIME * SENSe[1..n]:DATA:TELEcom:TEST:START: TIME:ENABled</pre>

:SENSe[1..n]:DATA:TELecom:TEST:STOP:TIME

Description	<p>This command sets a specific time the created test case will stop automatically.</p> <p>At *RST, this value is set to "1970-01-01 05:30:00".</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELecom:TEST:STOP:TIME <wsp> <Time></pre>
Parameter(s)	<p>Time: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets a specific time the created test case will stop automatically.</p>
Example(s)	<pre>* SENS:DATA:TEL:TEST:STOP:TIME "2007-09-10 05:30:00" * SENS:DATA:TEL:TEST:STOP:TIME? Returns "2007-09-10 05:30:00"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELecom:TEST:STOP:TIME? * SENSe[1..n]:DATA:TELecom:TEST:STOP:TIME: ENABLEd</pre>

:SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME?

Description	<p>This query returns the specific time the created test case will stop automatically.</p> <p>At *RST, this value is set to "1970-01-01 05:30:00".</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME?
Parameter(s)	None
Response Syntax	<Time>
Response(s)	<p>Time:</p> <p>The response data syntax for <Time> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the specific time the created test case will stop automatically.</p>
Example(s)	<p>* SENS:DATA:TEL:TEST:STOP:TIME "2007-09-10 05:30:00"</p> <p>* SENS:DATA:TEL:TEST:STOP:TIME?</p> <p>Returns "2007-09-10 05:30:00"</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME</p> <p>* SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME:ENABled</p>

**:SENSe[1..n]:DATA:TELEcom:TEST:STOP:
TIME:ENABled**

Description This command enables or disables the Stop Time configuration.

At *RST, this value is set to OFF.

Syntax :SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME:
ENABled<wsp><Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Stop Time configuration.

Example(s) * SENS:DATA:TEL:TEST:STOP:TIME "2007-09-10
05:30:00"
* SENS:DATA:TEL:TEST:STOP:TIME:ENAB ON
* SENS:DATA:TEL:TEST:STOP:TIME:ENAB?
Returns 1

See Also * SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME
* SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME:
ENABled?

**:SENSe[1..n]:DATA:TELEcom:TEST:STOP:
TIME:ENABLEd?**

Description	<p>This query returns the status of Stop Time configuration.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME: ENABLEd?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the Stop Time configuration.</p>
Example(s)	<pre>* SENS:DATA:TEL:TEST:STOP:TIME "2007-09-10 05:30:00" * SENS:DATA:TEL:TEST:STOP:TIME:ENAB ON * SENS:DATA:TEL:TEST:STOP:TIME:ENAB? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME * SENSe[1..n]:DATA:TELEcom:TEST:STOP:TIME: ENABLEd</pre>

**:SENSe[1..n]:DATA:TELEcom:TEST:DURation:
TIME**

Description	<p>This command selects the test duration time based on the test case start time.</p> <p>At *RST, this value is set to D15MIN.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME <wsp>D15MIN D1HOUR D2HOUR D24HOUR D48HOUR D72HOUR D7DAYS UDEfined</p>
Parameter(s)	<p>Time: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: D15MIN D1HOUR D2HOUR D24HOUR D48HOUR D72HOUR D7DAYS UDEfined. Selects the test duration based on the test case start time. D15MIN, select the test duration as 15 minutes. D1HOUR, select the test duration as 1 hour. D2HOUR, select the test duration as 2 hours. D24HOUR, select the test duration as 24 hours. D48HOUR, select the test duration as 48 hours. D72HOUR, select the test duration as 72 hours. D7DAYS, select the test duration as 7 days.</p>

**:SENSe[1..n]:DATA:TELEcom:TEST:DURation:
TIME**

UDEFined, select the test duration as user defined.

Example(s)

* SENS:DATA:TEL:TEST:DUR:TIME D2HOUR
* SENS:DATA:TEL:TEST:DUR:TIME?
Returns D2HOUR

See Also

* SENSe[1..n]:DATA:TELEcom:TEST:DURation:
TIME?
* SENSe[1..n]:DATA:TELEcom:TEST:DURation:
TIME:ENABled

:SENSe[1..n]:DATA:TELEcom:TEST:DURation:TIME?

Description	<p>This query returns the test duration time based on the test case start time.</p> <p>At *RST, this value is set to D15MIN.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:TEST:DURation:TIME?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Time></p>
Response(s)	<p>Time: The response data syntax for <Time> is defined as a <CHARACTER RESPONSE DATA> element. Returns the test duration based on the test case start time. D15MIN, test duration as 15 minutes is selected. D1HOUR, test duration as 1 hour is selected. D2HOUR, test duration as 2 hours is selected. D24HOUR, test duration as 24 hours is selected. D48HOUR, test duration as 48 hours is selected. D72HOUR, test duration as 72 hours is selected. D7DAYS, test duration as 7 days is selected.</p>

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

UNDEFINED, test duration as user defined is selected.

Example(s)

* SENS:DATA:TEL:TEST:DUR:TIME D2HOUR
* SENS:DATA:TEL:TEST:DUR:TIME?
Returns D2HOUR

See Also

* SENSe[1..n]:DATA:TELEcom:TEST:DURation:
TIME
* SENSe[1..n]:DATA:TELEcom:TEST:DURation:
TIME:ENABled

**:SENSe[1..n]:DATA:TELEcom:TEST:DURation:
TIME:ENABled**

Description	<p>This command enables or disables the test duration based on the test case start time.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME:ENABled <wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the test duration based on the test case start time.</p>
Example(s)	<p>* SENS:DATA:TEL:TEST:DUR:TIME D2HOUR * SENS:DATA:TEL:TEST:DUR:TIME:ENAB ON * SENS:DATA:TEL:TEST:DUR:TIME:ENAB? Returns 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME * SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME:ENABled?</p>

**:SENSe[1..n]:DATA:TELEcom:TEST:DURation:
TIME:ENABled?**

Description	<p>This query returns the status of the test duration based on the test case start time.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<code>:SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME:ENABled?</code>
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the test duration based on the test case start time.</p>
Example(s)	<pre>* SENS:DATA:TEL:TEST:DUR:TIME D2HOUR * SENS:DATA:TEL:TEST:DUR:TIME:ENAB ON * SENS:DATA:TEL:TEST:DUR:TIME:ENAB? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME * SENSe[1..n]:DATA:TELEcom:TEST:DURation: TIME:ENABled</pre>

**:SENSe[1..n]:DATA:TELEcom:TEST:DURation:
USER**

Description This command sets the test duration when User Defined has been selected for duration.

At *RST, this value is set to "00d:00:15:00".

Syntax :SENSe[1..n]:DATA:TELEcom:TEST:DURation:
USER<wsp><Duration>

Parameter(s) Duration:
The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.
Sets the test duration when User Defined has been selected for duration.

Example(s) * SENS:DATA:TEL:TEST:DUR:TIME DUD
* SENS:DATA:TEL:TEST:DUR:USER "00d:00:15:00"
* SENS:DATA:TEL:TEST:DUR:USER?
Returns "00d:00:15:00"

See Also * SENSe[1..n]:DATA:TELEcom:TEST:DURation:
TIME
* SENSe[1..n]:DATA:TELEcom:TEST:DURation:
TIME:ENABled
* SENSe[1..n]:DATA:TELEcom:TEST:DURation:
USER?

**:SENSe[1..n]:DATA:TELEcom:TEST:DURation:
USER?**

Description	This query returns the test duration when User Defined has been selected for duration. At *RST, this value is set to "00d:00:15:00".
Syntax	:SENSe[1..n]:DATA:TELEcom:TEST:DURation: USER?
Parameter(s)	None
Response Syntax	<Duration>

:SENSe[1..n]:DATA:TELEcom:TEST:DURation: USER?

Response(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the test duration when User Defined has been selected for duration.</p>
Example(s)	<p>* SENS:DATA:TEL:TEST:DUR:TIME DUD</p> <p>* SENS:DATA:TEL:TEST:DUR:USER "00d:00:15:00"</p> <p>* SENS:DATA:TEL:TEST:DUR:USER?</p> <p>Returns "00d:00:15:00"</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:TEST:DURation:TIME</p> <p>* SENSe[1..n]:DATA:TELEcom:TEST:DURation:TIME:ENABled</p> <p>* SENSe[1..n]:DATA:TELEcom:TEST:DURation:USER</p>

:SENSe[1..n]:DATA:TELEcom:TEST:TIMer:RESet

Description	<p>This command enables or disables the test timer.</p> <p>This command is an event and has no associated *RST condition.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:TEST:TIMer:RESet
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:TEST:DUR:TIME DUD* SENS:DATA:TEL:TEST:DUR:USER "00d:00:15:00"* SENS:DATA:TEL:TEST:TIMer:RESet* SENS:DATA:TEL:TEST:TIMer:STATus? Returns 1
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:TEST:DURation:TIME* SENSe[1..n]:DATA:TELEcom:TEST:DURation:TIME:ENABled* SENSe[1..n]:DATA:TELEcom:TEST:TIMer:RESet

:SENSe[1..n]:DATA:TELEcom:TEST:TIMer:STATUs?

Description	This query returns the status of test timer. At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:TEST:TIMer:STATUs?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of test timer.
Example(s)	* SENS:DATA:TEL:TEST:DUR:TIME DUD * SENS:DATA:TEL:TEST:DUR:USER "00d:00:15:00" * SENS:DATA:TEL:TEST:TIMer:RESet * SENS:DATA:TEL:TEST:TIME:STATUs? Returns 1
See Also	* SENSe[1..n]:DATA:TELEcom:TEST:DURation:TIME * SENSe[1..n]:DATA:TELEcom:TEST:TIMer:RESet * SENSe[1..n]:DATA:TELEcom:TEST:TIMer:STATUs?

:SOURCE[1..n]:DATA:TELEcom:COUPled

Description	<p>This command enables or disables the transmitter coupling.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:COUPled<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the transmitter coupling.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:COUP ON* SOUR:DATA:TEL:COUP? Returns 1
See Also	<ul style="list-style-type: none">* SOURCE[1..n]:DATA:TELEcom:COUPled?* SOURCE[1..n]:DATA:TELEcom:THROUGH* SOURCE[1..n]:DATA:TELEcom:INTRusive

:SOURce[1..n]:DATA:TELEcom:COUPled?

Description	This query returns the status of transmitter coupling. At *RST, this value is set to ON.
Syntax	:SOURce[1..n]:DATA:TELEcom:COUPled?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of transmitter coupling.
Example(s)	* SOUR:DATA:TEL:COUP ON * SOUR:DATA:TEL:COUP? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:COUPled * SOURce[1..n]:DATA:TELEcom:THROUGH * SOURce[1..n]:DATA:TELEcom:INTRusive

:SOURce[1..n]:DATA:TELEcom:THROugh

Description	This command enables or disables the Through mode. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:THROugh <wsp> <Set>
Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Through mode.
Example(s)	* SOUR:DATA:TEL:COUP ON * SOUR:DATA:TEL:THR ON * SOUR:DATA:TEL:THR? Returns 1
Note	Through is available when Coupled is selected.
See Also	* SOURce[1..n]:DATA:TELEcom:COUPled * SOURce[1..n]:DATA:TELEcom:THROugh? * SOURce[1..n]:DATA:TELEcom:INTRusive

:SOURce[1..n]:DATA:TELEcom:THRough?

Description	This query returns the status of Through mode. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:THRough?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Through mode.
Example(s)	* SOUR:DATA:TEL:COUP ON * SOUR:DATA:TEL:THR ON * SOUR:DATA:TEL:THR? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:COUPled * SOURce[1..n]:DATA:TELEcom:THRough * SOURce[1..n]:DATA:TELEcom:INTRusive

:SOURce[1..n]:DATA:TELEcom:INTRusive

Description	<p>This command enables or disables the Intrusive mode.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:INTRusive <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Intrusive mode.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:COUP ON* SOUR:DATA:TEL:THR ON* SOUR:DATA:TEL:INTR ON* SOUR:DATA:TEL:INTR? Returns 1
Note	<p>Intrusive is available when Through is selected.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:COUPled* SOURce[1..n]:DATA:TELEcom:THROugh* SOURce[1..n]:DATA:TELEcom:INTRusive?

:SOURce[1..n]:DATA:TELEcom:INTRusive?

Description	This query returns the status of the Intrusive mode. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:INTRusive?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Intrusive mode.
Example(s)	* SOUR:DATA:TEL:COUP ON * SOUR:DATA:TEL:THR ON * SOUR:DATA:TEL:INTR ON * SOUR:DATA:TEL:INTR? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:COUPled * SOURce[1..n]:DATA:TELEcom:THRough * SOURce[1..n]:DATA:TELEcom:INTRusive

:SOURce[1..n]:DATA:TELEcom:INTRusive:OTN

Description	<p>This command enables or disables the OTN Intrusive mode.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:INTRusive:OTN <wsp><Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the OTN Intrusive mode.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:COUP ON* SOUR:DATA:TEL:THR ON* SOUR:DATA:TEL:INTR:OTN ON* SOUR:DATA:TEL:INTR:OTN? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:COUPled* SOURce[1..n]:DATA:TELEcom:THROUGH* SOURce[1..n]:DATA:TELEcom:INTRusive:OTN?

:SOURce[1..n]:DATA:TELEcom:INTRusive:OTN?

Description This query returns the status of the OTN Intrusive mode.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:INTRusive:OTN?

Parameter(s) None

Response Syntax <Set>

Response(s) Set:
The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the status of the OTN Intrusive mode.

Example(s)

- * SOUR:DATA:TEL:COUP ON
- * SOUR:DATA:TEL:THR ON
- * SOUR:DATA:TEL:INTR:OTN ON
- * SOUR:DATA:TEL:INTR:OTN? Returns 1

See Also

- * SOURce[1..n]:DATA:TELEcom:COUPled
- * SOURce[1..n]:DATA:TELEcom:THRough
- * SOURce[1..n]:DATA:TELEcom:INTRusive:OTN

:SOURce[1..n]:DATA:TELEcom:TSL**Description**

This command is used to change the Dynamic Time Slot value for different test data paths.

There are 6 parameters in all. The first and the second parameters are mandatory. The remaining four are optional, as these parameters depend upon the dynamic time slot value to be set.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:TSL<wsp>HOP |
LOP, <LEVEL1>, LEVEL2>

:SOURCE[1..n]:DATA:TELEcom:TSL

Parameter(s)

Type:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: HOP|LOP.

Selects the TSL type.

HOP, selects the High Order Path as the TSL type.

LOP, selects the Low Order Path as the TSL type.

LEVEL1:

The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the Dynamic time slot value for LEVEL1.

LEVEL2:

The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the Dynamic time slot value for LEVEL2.

LEVEL3:

The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the Dynamic time slot value for LEVEL3.

:SOURce[1..n]:DATA:TELEcom:TSL

Example(s) SOUR:DATA:TEL:COUP ON
 SOUR:DATA:TEL:THR ON
 SOUR:DATA:TEL:INTR:OTN ON
 SOUR:DATA:TEL:INTR:OTN? Returns 1

Note Intrusive is available when Through is selected.

See Also SOURce[1..n]:DATA:TELEcom:COUPled
 SOURce[1..n]:DATA:TELEcom:THROUGH
 SOURce[1..n]:DATA:TELEcom:INTRusive:OTN?

:SOURce[1..n]:DATA:TELEcom:TSL?

Description	<p>This query returns the Dynamic Time Slot value for different test data paths.</p> <p>There are 6 parameters in all, aout of which the first and the second paramaters are mandatory, and the rest are not, as these parameters depend upon the dynamic time slot value to be set.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:TSL?<wsp>HOP LOP</pre>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: HOP LOP.</p> <p>Selects the TSL type.</p> <p>HOP, selects the High Order Path as the TSL type.</p> <p>LOP, selects the Low Order Path as the TSL type.</p>
Response Syntax	<pre><Slots></pre>

:SOURce[1..n]:DATA:TELEcom:TSL?**Response(s)**

Slots:

The response data syntax for <Slots> is defined as a <STRING RESPONSE DATA> element.

Returns the selected TSL type.

Example(s)

SOUR:DATA:TEL:COUP ON

SOUR:DATA:TEL:THR ON

SOUR:DATA:TEL:INTR:OTN ON

SOUR:DATA:TEL:INTR:OTN? Returns 1”

See Also

SOURce[1..n]:DATA:TELEcom:COUPled

SOURce[1..n]:DATA:TELEcom:THRough

SOURce[1..n]:DATA:TELEcom:INTRusive:OTN

:SOURce[1..n]:DATA:TELEcom:TEST:NODE: DIRection

Description

This command sets the direction for configuring the test.

To configure tests separately on both the direction we need to use this command.

Example: To configure the Test on TX, we will execute this command with parameter as TX, then execute the expected configuration commands, which will configure a Test at TX-direction. Similar Steps needs to be followed to configure a separate test at RX-direction.

This command is not associated with any *RST condition.

Syntax

:SOURce[1..n]:DATA:TELEcom:TEST:NODE:
DIRection <wsp> TX|RX

:SOURce[1..n]:DATA:TELEcom:TEST:NODE: DIRection

Parameter(s)	<p>Node Direction:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TX RX</p> <p>Sets the direction for configuring the test.</p> <p>TX, sets TX as the configuration direction.</p> <p>RX, sets RX as the configuration direction.</p>
Example(s)	<pre>SOUR:DATA:TEL:COUP OFF SOURce:DATA:TELEcom:TEST:NODE:DIRection TX SOURce:DATA:TELEcom:TEST:NODE:DIRection? OUTPut:TELEcom:CONNector OPTical SOURce:DATA:TELEcom:INTerface:TYPE OTU2 SOURce:DATA:TELEcom:ODU:TYPE O2O1ODU0OC12 SOURce:DATA:TELEcom:HOP:TYPE STS3C SOURce:DATA:TELEcom:TEST:NODE:DIRection RX SOURce:DATA:TELEcom:TEST:NODE:DIRection? OUTPut:TELEcom:CONNector OPTical SOURce:DATA:TELEcom:INTerface:TYPE OTU1 SOURce:DATA:TELEcom:ODU:TYPE O1ODU0OC12 SOURce:DATA:TELEcom:HOP:TYPE STS12C</pre>
See Also	<pre>SOUR:DATA:TEL:THR ON</pre>

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

Description	<p>This query returns the direction for which the test is been configured.</p> <p>This query is not associated with any *RST condition.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:TEST:NODE:DIRection?
Parameter(s)	None
Response Syntax	<Direction>

:SOURce[1..n]:DATA:TELEcom:TEST:NODE: DIRection?

Response(s)	<p>Direction:</p> <p>The response data syntax for <Direction> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the selected direction.</p> <p>TX, returns TX as the configuration direction.</p> <p>RX, returns RX as the configuration direction.</p>
Example(s)	<pre> SOUR:DATA:TEL:COUP OFF SOURce:DATA:TELEcom:TEST:NODE:DIRection TX SOURce:DATA:TELEcom:TEST:NODE:DIRection? OUTPut:TELEcom:CONNector OPTical SOURce:DATA:TELEcom:INTerface:TYPE OTU2 SOURce:DATA:TELEcom:ODU:TYPE O2O1ODU0OC12 SOURce:DATA:TELEcom:HOP:TYPE STS3C SOURce:DATA:TELEcom:TEST:NODE:DIRection RX SOURce:DATA:TELEcom:TEST:NODE:DIRection? OUTPut:TELEcom:CONNector OPTical SOURce:DATA:TELEcom:INTerface:TYPE OTU1 SOURce:DATA:TELEcom:ODU:TYPE O1ODU0OC12 SOURce:DATA:TELEcom:HOP:TYPE STS12C </pre>
See Also	<pre>SOUR:DATA:TEL:INTR?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:TRIButary: PORT

Description

This Command sets the Tributary port for the selected ODU Layer.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:TRIButary:  
PORT <wsp>ODU1|ODU2|ODU1F|ODU0|  
ODU2e|ODU2f|ODU3|O2ODUFIEX,<PORT  
NUMBER>
```

:SOURce[1..n]:DATA:TELEcom:OTN:TRIButary: PORT

Parameter(s)

ODU Layer:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

ODU1|ODU2|ODU1F|ODU0|ODU2e|ODU2f|
ODU3|O2ODUFIEX <PORT NUMBER>

Selects the ODU Layer for configuring the Tributary Port

ODU1

ODU2

ODU1F

ODU0

ODU2e

ODU2f

ODU3

O2ODUFIEX

PORT NUMBER: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

The allowed <DECIMAL NUMERIC PROGRAM DATA> elements for this parameter are:

Sets the Tributary Port values

Response Syntax

**:SOURce[1..n]:DATA:TELEcom:OTN:TRIButary:
PORT**

Response(s)

Example(s)

SOURce:DATA:TELEcom:OTN:TRIButary:PORT
o2ODUFLEX,3
SOURce:DATA:TELEcom:OTN:TRIButary:PORT
ODUFLEX,6
SOURce:DATA:TELEcom:OTN:TRIButary:PORT
ODU1,2

See Also

SOURce:DATA:TELEcom:TEST:NODE:DIRection
TX
SOURce:DATA:TELEcom:TEST:NODE:DIRection?
OUTPut:TELEcom:CONNector?
OUTPut:TELEcom:CONNector OPTical
OUTPut:TELEcom:CONNector?
SOURce:DATA:TELEcom:INTerface:TYPE?
SOURce:DATA:TELEcom:INTerface:TYPE OTU2

:SOURce[1..n]:DATA:TELEcom:OTN:TRIButary:PORT?

Description	<p>This Query gets the Tributary Port for selected ODU layer.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:TRIButary:PORT?<wsp> ODU1 ODU2 ODU1F ODU0 ODU2e ODU2f ODU3 O2ODUFIEX
Parameter(s)	<p>ODU Layer:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: ODU1 ODU2 ODU1F ODU0 ODU2e ODU2f ODU3 O2ODUFIEX</p> <p>Selects the ODU Layer for configuring the Tributary Port</p> <p>ODU1 ODU2 ODU1F ODU0 ODU2e ODU2f ODU3 O2ODUFIEX</p>

:SOURce[1..n]:DATA:TELEcom:OTN:TRIButary:PORT?

Response Syntax <Set>

Response(s) Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the Tributary Port value.

Example(s) SOURce:DATA:TELEcom:OTN:TRIButary:PORT?
ODUFLEX
SOURce:DATA:TELEcom:OTN:TRIButary:PORT?
ODU1

See Also SOURce:DATA:TELEcom:TEST:NODE:DIRection
TX
SOURce:DATA:TELEcom:TEST:NODE:DIRection?
OUTPut:TELEcom:CONNector?
OUTPut:TELEcom:CONNector OPTical
OUTPut:TELEcom:CONNector?
SOURce:DATA:TELEcom:INTerface:TYPE?
SOURce:DATA:TELEcom:INTerface:TYPE OTU2
SOURce:DATA:TELEcom:INTerface:TYPE?

Clock Synchronization Command Reference

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

Description

This command sets the external timing line code for the output port.

At *RST, this value is set to B8ZS.

Syntax

:OUTPut[1..n]:TELEcom:CODE<wsp>B8ZS |
HDB3 |AMI

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

Parameter(s)	<p>Code:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>B8ZS HDB3 AMI.</p> <p>Sets the external timing line code.</p> <p>B8ZS, selects B8ZS as external timing line code.</p> <p>HDB3, selects HDB3 as external timing line code.</p> <p>AMI, selects AMI as external timing line code.</p>
Example(s)	<ul style="list-style-type: none">* OUTP:TEL:LEV DS1L* OUTP:TEL:CODE HDB3* OUTP:TEL:CODE? Returns HDB3
See Also	<ul style="list-style-type: none">* OUTPut[1..n]:TELEcom:CODE?

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

Description This query returns the external timing line code for the output port.

At *RST, this value is set to B8ZS.

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

Parameter(s) None

Response Syntax <Code>

:OUTPut[1..n]:TELecom:CODE?

Response(s)

Code:

The response data syntax for <Code> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the external timing line code.

B8ZS1, B8ZS external timing line code is selected.

HDB3, HDB3 external timing line code is selected.

AMI, AMI external timing line code is selected.

Example(s)

* OUTP:TEL:LEV E1

* OUTP:TEL:CODE HDB3

* OUTP:TEL:CODE? Returns HDB3

See Also

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

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

This command selects the interface framing.

At *RST, this value is set to SF.

Syntax

:OUTPut[1..n]:TELEcom:FRAMing<wsp>SF |
ESF1 | PCM30 | PCM30C4 | PCM31 | PCM31C4

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

Parameter(s)	<p>Framing:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>SF ESF1 PCM30 PCM30C4 PCM31 PCM31C4.</p> <p>Sest the interface framing.</p> <p>Selects the interface framing.</p> <p>SF, selects the SF (Superframe) as interface framing</p> <p>ESF, selects the ESF (Extended Superframe) as interface framing.</p> <p>PCM30, selects the PCM30 (Pulse Code Modulation) as interface framing.</p> <p>PCM30C4, selects the PCM30 (Pulse Code Modulation) CRC (Cyclic Redundancy Check) as interface framing.</p> <p>PCM31 selects the PCM31 (Pulse Code Modulation) as interface framing.</p> <p>PCM31C4, selects the PCM31(Pulse Code Modulation) CRC (Cyclic Redundancy Check) as interface framing</p>
---------------------	---

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

Example(s)

- * OUTP:TEL:LEV DS1LEVEL
- * OUTP:TEL:FRAM ESF1
- * OUTP:TEL:FRAM? Returns ESF1

See Also

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

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

Description This query returns the interface framing.

At *RST, this value is set to SF.

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

Parameter(s) None

Response Syntax <Framing>

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

Response(s)	<p>Framing:</p> <p>The response data syntax for <Framing> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the interface framing.</p> <p>SF, Superframe (SF) is selected.</p> <p>ESF, Extended Superframe (ESF) is selected.</p> <p>PCM30, Pulse Code Modulation (PCM30) is selected.</p> <p>PCM30C4, Pulse Code Modulation (PCM30) Cyclic Redundancy Check (CRC) is selected.</p> <p>PCM31, Pulse Code Modulation (PCM31) is selected.</p> <p>PCM31C-4, Pulse Code Modulation (PCM31) Cyclic Redundancy Check (CRC) is selected.</p>
Example(s)	<ul style="list-style-type: none">* OUTP:TEL:LEV DS1LEVEL* OUTP:TEL:FRAM ESF1* OUTP:TEL:FRAM? Returns ESF1
See Also	<ul style="list-style-type: none">* OUTPut[1..n]:TELEcom:FRAMing

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

Description This command sets the external timing interface output level for the output port.

At *RST, this value is set to NONE.

Syntax :OUTPut[1..n]:TELEcom:LEVel<wsp>NONE |
DS1L|E1BNC|L2MHZ

:OUTPut[1..n]:TELEcom:LEVel**Parameter(s)**

Level:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

NONE|DS1L|E1BNC|L2MHZ.

Selects the external timing interface output level.

NONE, No external timing interface output level is selected.

DS1, selects DS1 (Digital Signal-level 1) as external timing interface output level.

E1BNC, selects the E1BNC as the external timing interface output level.

L2MHZ, selects the L2MHZ as the external timing interface output level.

Example(s)

* OUTP:TEL:LEV DS1LEVEL

* OUTP:TEL:LEV? Returns DS1LEVEL

See Also

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

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

Description This query returns the external timing interface output level for the output port.

At *RST, this value is set to NONE.

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

Parameter(s) None

Response Syntax <Level>

:OUTPut[1..n]:TELEcom:LEVel?**Response(s)**

Level:

The response data syntax for <Level> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the external timing interface output level.

NONE, No external timing interface output level is selected.

DS1, Digital Signal-level 1 (DS1) as external timing interface output level is selected.

E1BNC, E1BNC as the external timing interface output level is selected.

L2MHZ, L2MHZ as the external timing interface output level is selected.

Example(s)

* OUTP:TEL:LEV DS1L

* OUTP:TEL:LEV? Returns DS1LEVEL

See Also

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

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

Description This command sets the termination mode for the external timing.

At *RST, this value is set to TERM.

Syntax :OUTPut[1..n]:TELEcom:TERMination <wsp>
TERM|MON|BRIDge

Parameter(s) Termination:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> element for this parameter is: TERM.
Sets the termination mode.
TERM, selects the Term as termination mode.
MON, selects the Mon as termination mode.
BRIDge, selects the Bridge as termination mode.

Example(s) *OUTP:TEL:LEV DS ILEVEL
*OUTP:TEL:TERM TERM
*OUTP:TEL:TERM? Returns:TERM

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

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

Description	This query returns the termination mode. At *RST, this value is set to TERM.
Syntax	:OUTPut[1..n]:TELEcom:TERMination?
Parameter(s)	None
Response Syntax	<Termination>
Response(s)	Termination: The response data syntax for <Termination> is defined as a <CHARACTER RESPONSE DATA> element. Returns the termination mode. TERM, Term as termination mode is selected. MON, Mon as termination mode is selected. BRIDge, Bridge as termination mode is selected.
Example(s)	* OUTP:TEL:LEV DS1L * OUTP:TEL:TERM TERM * OUTP:TEL:TERM? Returns TERM
See Also	* OUTPut[1..n]:TELEcom:TERMination

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

Description	<p>This query returns the clock alarm status for the output port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:OUTPut[1..n]:TELEcom:CLOCK:ALARm: STATus? <wsp>LOS AIS LOF FREQUENCY</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOS AIS LOF FREQUENCY.</p> <p>Sets the clock alarm status.</p> <p>LOS, selects LOS (Loss Of Signal) as the alarm status.</p> <p>AIS, selects AIS (Alarm Indication Signal) as the alarm status.</p> <p>LOF, selects LOF(Loss Of Frequency) as the alarm status.</p> <p>FREQUENCY, selects frequency as the clock alarm status.</p>

**:OUTPut[1..n]:TELEcom:CLOCK:ALARm:
STATus?****Response Syntax** <Status>**Response(s)**

Status:

The response data syntax for <Status> is defined as <CHARACTER RESPONSE DATA> element.

Returns the clock alarm status.

PRESENT, indicates that at least one alarm/error has occurred.

ABSENT, indicates that no alarm/error occurred.

INACTIVE, indicates that the test did not run yet.

Example(s)

* OUTP:TEL:LEV DS1LEVEL

* OUTP:TEL:CLOC:ALAR:STAT? LOS Returns the clock alarm status.t.

:OUTPut[1..n]:TELEcom:CLOCK:FREQUency?

Description	This query returns the received signal rate. At *RST, this value is device dependent.
Syntax	:OUTPut[1..n]:TELEcom:CLOCK:FREQUency?
Parameter(s)	None
Response Syntax	<Frequency>
Response(s)	Frequency: The response data syntax for <Frequency> is defined as <NR2 NUMERIC RESPONSE DATA> element. Returns the received signal rate.
Example(s)	* OUTP:TEL:LEV DS1LEVEL * OUTP:TEL:CLOC:FREQ? Returns the frequency of the input signal for output signal.
See Also	* OUTPut[1..n]:TELEcom:CLOCK:FREQUency:OFFSet?

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

Description	<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, this value is device dependent.</p>
Syntax	:OUTPut[1..n]:TELEcom:CLOCK:FREQuency: OFFSet?
Parameter(s)	None
Response Syntax	<Offset>
Response(s)	<p>Offset:</p> <p>The response data syntax for <Offset> is defined as <STRING RESPONSE DATA> element.</p> <p>Returns the frequency offset.</p>
Example(s)	<p>* OUTP:TEL:LEV DS1LEVEL</p> <p>* OUTP:TEL:CLOC:FREQ:OFFS? Returns the offset value of the frequency.</p>
See Also	* OUTPut[1..n]:TELEcom:CLOCK:FREQuency?

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

Description This command sets the interface line coding for the input port.

At *RST, this value is set to B8ZS.

Syntax :INPut[1..n]:TELEcom:CODE<wsp>B8ZS|HDB3|AMI

Parameter(s) Code:
The program data syntax for <Code> is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
B8ZS|HDB3|AMI.
Sets the interface line coding.
B8ZS, selects the B8ZS as interface line coding.
HDB3, selects the HDB3 as interface line coding.
AMI, selects the AMI as interface line coding.

Example(s) *INP:TEL:LEV E1
*INP:TEL:CODE HDB3
*INP:TEL:CODE? Returns:HDB3

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

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

Description	<p>This query returns the interface line coding for the input port.</p> <p>At *RST, this value is set to B8ZS.</p>
Syntax	:INPut[1..n]:TELEcom:CODE? <wsp>
Parameter(s)	None
Response Syntax	<Code>
Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as <STRING RESPONSE DATA> element.</p> <p>Returns the interface line coding.</p> <p>B8ZS is selected as the interface line coding.</p> <p>HDB3 is selected as the interface line coding.</p> <p>AMI is selected as the interface line coding.</p>
Example(s)	<ul style="list-style-type: none">* INP:TEL:LEV E1* INP:TEL:CODE HDB3* INP:TEL:CODE? Returns:HDB3
See Also	* INPut[1..n]:TELEcom:CODE.

:INPut[1..n]:TELecom:FRAMing

Description

This command selects the interface framing.

At *RST, this value is set to SF.

Syntax

:INPut[1..n]:TELecom:FRAMing<wsp>SF|ESF1|
PCM30|PCM30CRC-4|PCM31|PCM31CRC-4

:INPut[1..n]:TELEcom:FRAMing**Parameter(s)**

Framing:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

SF|ESF1|PCM30|PCM30CRC-4|PCM31|PCM31CRC-4.

Sets the interface framing.

SF, selects the SF (Superframe) as interface framing.

ESF1, selects the ESF (Extended Superframe) as interface framing.

PCM30, selects the PCM30 (Pulse Code Modulation) as interface framing.

PCM30CRC, selects the PCM30 (Pulse Code Modulation) CRC (Cyclic Redundancy Check) as interface framing.

PCM31 selects the PCM31 (Pulse Code Modulation) as interface framing.

PCM31CRC, selects the PCM31 (Pulse Code Modulation) CRC (Cyclic Redundancy Check) as interface framing.

Example(s)

* INP:TEL:LEV DS1LEVEL

* INP:TEL:FRAM ESF1

* INP:TEL:FRAM? Returns:ESF1

See Also

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

:INPut[1..n]:TELecom:FRAMing?

Description This query returns the interface framing.

At *RST, this value is set to SF.

Syntax :INPut[1..n]:TELecom:FRAMing?<wsp>

Parameter(s) None

Response Syntax <Framing>

:INPut[1..n]:TELEcom:FRAMing?**Response(s)**

Framing:

The response data syntax for <Framing> is defined as a <CHARACTER RESPONSE DATA> RESPONSE DATA> element.

Returns the interface framing.

SF, SF (Superframe) is selected as the interface framing.

ESF1, ESF (Extended Superframe) is selected as the interface framing.

PCM30, PCM30 (Pulse Code Modulation) is selected as the interface framing.

PCM30CRC, PCM30 (Pulse Code Modulation) CRC (Cyclic Redundancy Check) is selected as the interface framing.

PCM31, PCM31 (Pulse Code Modulation) as interface framing.

PCM31CRC, PCM31 (Pulse Code Modulation) CRC (Cyclic Redundancy Check) is selected as the interface framing.

Example(s)

* INP:TEL:LEV DS1LEVEL

* INP:TEL:FRAM ESF1

* INP:TEL:FRAM? Returns:ESF1

See Also

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

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

Description This command sets the internal timing interface input level for the input port.

At *RST, this value is set to NONE.

Syntax :INPut[1..n]:TELEcom:LEVel<wsp>NONE|
DS1L|E1BNC|L2MHZ

Parameter(s) Level:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
NONE|DS1L|E1BNC|L2MHZ.
Sets the internal timing interface input level.
NONE, No interface type is selected.
DS1Level, DS1 (Digital Signal-level 1) as the interface type is selected.
E1, E1 as the interface type is selected.
L2MHZ, L2MHZ as the interface type is selected.

Example(s) * INP:TEL:LEV DS1LEVEL
* INP:TEL:LEV? Returns:DS1LEVEL

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

:INPut[1..n]:TELecom:LEVel?

Description This query returns the internal timing interface input level for the input port.

At *RST, this value is set to NONE.

Syntax :INPut[1..n]:TELecom:LEVel?<wsp>

Parameter(s) None

Response Syntax <Level>

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

Response(s)

Level:

The response data syntax for <Level> is defined as a <CHARACTER RESPONSE DATA> RESPONSE DATA> element.

Returns the internal timing interface input level.

NONE, No interface type is selected.

DS1LEVEL, DS1 (Digital Signal-level 1) as the interface type is selected.

E1, E1 as the interface type is selected.

L2MHZ, L2MHZ as the interface type is selected.

Example(s)

* INP:TEL:LEV DS1LEVEL

* INP:TEL:LEV? Returns:DS1LEVEL

See Also

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

:INPut[1..n]:TELecom:CLOCK**Description**

This command sets the configuration of the clock that will be generated.

At *RST, this value is set to INTERNAL.

Syntax

:INPut[1..n]:TELecom:CLOCK<wsp>INTernal |
EXTernal | RECovered | BPLane

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

Parameter(s)	<p>Clock:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: INTErnal EXTernal RECOVered BPLane.</p> <p>Sets the configuration of the clock that will be generated.</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>RECOVered, indicates the live clock from the optical/electrical port input signal involved in the test.</p> <p>BPLane, indicates the 8 kHz clock from another module on the FTB-400.</p>
Example(s)	<ul style="list-style-type: none">* INP:TEL:LEV DS1LEVEL* INP:TEL:CLOC REC* INP:TEL:CLOC? Returns: RECOVERED
See Also	<ul style="list-style-type: none">* INPut[1..n]:TELEcom:CLOCK?

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

Description This query returns the configuration of the clock that is generated.

At *RST, this value is set to INTERNAL.

Syntax :INPut[1..n]:TELEcom:CLOCK? <wsp>

Parameter(s) None

Response Syntax <Clock>

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

Response(s)

Clock:

The response data syntax for <Clock> is defined as a <CHARACTER RESPONSE DATA> RESPONSE DATA> element.

Returns the configuration of the clock that is generated.

INTERNAL, the internal clock of the unit (STRATUM 3) is indicated.

EXTERNAL, the clock received from the connected DS1/E1/2M external clock signal (port) is indicated.

RECOVERED, the live clock from the optical/electrical port input signal involved in the test is indicated.

BPLANE, the 8 kHz clock from another module on the FTB-400 is indicated.

Example(s)

* INP:TEL:LEV DS1LEVEL

* INP:TEL:CLOC REC

* INP:TEL:CLOC? Returns: RECOVERED

See Also

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

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

This command sets the value for the Line Build Out interface.

At *RST, this value is set to P30.

Syntax

:INPut[1..n]:TELEcom:LBO<wsp>P30|P24|P18|P12|P06

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

Parameter(s)	<p>Lbo:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: P30 P24 P18 P12 P06.</p> <p>Sets the Line Build Out interface value.</p> <p>P30, selects P30 as the Line Build Out interface value.</p> <p>P24, selects P24 as the Line Build Out interface value.</p> <p>P18, selects P18 as the Line Build Out interface value.</p> <p>P12, selects P12 as the Line Build Out interface value.</p> <p>P06, selects P06 as the Line Build Out interface value.</p>
Example(s)	<ul style="list-style-type: none">* INP:TEL:LEV DS1LEVEL* INP:TEL:LBO P30* INP:TEL:LBO? Returns P30
See Also	<ul style="list-style-type: none">* INPut[1..n]:TELEcom:LBO?

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

Description This query returns the value for the Line Build Out interface.

At *RST, this value is set to P30.

Syntax :INPut[1..n]:TELEcom:LBO?<wsp>

Parameter(s) None

Response Syntax <Clock>

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

Response(s)

Clock:

The response data syntax for <Clock> is defined as a <CHARACTER RESPONSE DATA> RESPONSE DATA> element.

Returns the value for the Line Build Out interface.

P30, means +3.0 dBdsx (533-655 ft).

P24, means +2.4 dBdsx (399-533 ft).

P18, means +1.8 dBdsx (266-399 ft).

P12, means +1.2 dBdsx (133-266 ft).

P06, means +0.6 dBdsx (0-133 ft).

Example(s)

* INP:TEL:LEV DS1LEVEL

* INP:TEL:LBO P30

* INP:TEL:LBO? Returns P30

See Also

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

:INPut[1..n]:TELEcom:CLOCK:ALARm:STATUS?

Description	<p>This query returns the status whether the module is able to synchronize with the selected clock mode or not.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:INPut[1..n]:TELEcom:CLOCK:ALARm:STATUS? <wsp>LOC</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOC.</p> <p>Sets the status of the clock synchronization.</p> <p>LOC, selects the LOC (Loss Of Clock) as the clock alarm status for the input port.</p>
Response Syntax	<pre><Status></pre>

:INPut[1..n]:TELEcom:CLOCK:ALARm:STATus?

Response(s)

Status:

The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> RESPONSE DATA> element.

Returns the status of the clock synchronization.

PRESENT, indicates the module is able to synchronize with the selected test clock.

ABSENT, indicates the module is not able to synchronize with the selected test clock.

Example(s)

* INP:TEL:LEV DS1LEVEL

* INP:TEL:CLOC:ALAR:STAT? LOC Returns the status of the clock synchronization.

See Also

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

:INPut[1..n]:TELEcom:CLOCK:SIGNal:STATus?

Description	<p>This query returns the status of a signal at the input interface/port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:INPut[1..n]:TELEcom:CLOCK:SIGNal:STATus? <wsp>OUTPut</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter is: OUTPut.</p> <p>Sets the status of a signal.</p> <p>OUTPut, selects output as the status of the signal.</p>
Response Syntax	<pre><Status></pre>

:INPut[1..n]:TELEcom:CLOCK:SIGNAL:STATus?

Response(s)

Status:

The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> RESPONSE DATA> element.

Returns the status of a signal.

PRESENT, indicates the signal is present.

ABSENT, indicates the signal is absent.

Example(s)

* INP:TEL:LEV DS1LEVEL

* INP:TEL:CLOC:SING:STAT? OUTP Returns the status of a signal.

See Also

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

:INPut[1..n]:TELeom:BACKplane:CLOCK**Description**

This command sets the backplane 8 kHz clock mode for synchronization at the input port.

At *RST, this value is set to INTERNAL.

Syntax

:INPut[1..n]:TELeom:BACKplane:CLOCK
<wsp>INTernal|EXTernal|RECovered

:INPut[1..n]:TELecom:BACKplane:CLOCK

Parameter(s)	<p>Clock:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>INTernal EXTernal RECovered</p> <p>Sets the backplane 8 kHz clock mode for synchronization.</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>RECovered, indicates the live clock from the optical/electrical port input signal involved in the test.</p>
Example(s)	<ul style="list-style-type: none">* INP:TEL:BACK:CLOC REC* INP:TEL:BACK:CLOC? Returns: RECOVERED
See Also	<ul style="list-style-type: none">* INPut[1..n]:TELecom:BACKplane:CLOCK?

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

Description This query returns the backplane 8 kHz clock mode at the input port.

At *RST, this value is set to INTERNAL.

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

Parameter(s) None

Response Syntax <Status>

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

Response(s)	<p>Clock:</p> <p>The response data syntax for <Status> is defined as <CHARACTER RESPONSE DATA> element.</p> <p>Returns the backplane 8 kHz clock mode.</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>RECOVERED, indicates the live clock from the optical/electrical port input signal involved in the test.</p>
Example(s)	<ul style="list-style-type: none">* INP:TEL:BACK:CLOC REC* INP:TEL:BACK:CLOC? Returns: RECOVERED
See Also	<ul style="list-style-type: none">* INPut[1..n]:TELEcom:BACKplane:CLOCK

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

This command sets the configuration of the backplane 8 kHz clock that will be generated when enabled.

At *RST, this value is set to OFF.

Syntax

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

Parameter(s)

Set:

The program data syntax for the parameter is defined as a <Boolean Program Data> element.

Enables or disables the configuration of the backplane 8 kHz clock.

Example(s)

* INP:TEL:BACK ON

* INP:TEL:BACK? Returns: 1

See Also

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

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

Description This query returns the status of the backplane 8 kHz clock synchronization.

At *RST, this value is set to OFF.

Syntax :INPut[1..n]:TELEcom:BACKplane?

Parameter(s) None

Response Syntax <Set>

Response(s) Set:
The response data syntax for <Set> is defined as <NR1 NUMERIC RESPONSE DATA> element.
Returns the status of the backplane 8 kHz clock synchronization.

Example(s) * INP:TEL:BACK ON
* INP:TEL:BACK? Returns: 1

See Also * INPut[1..n]:TELEcom:BACKplane

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

Description	<p>This query returns the status whether the module is able to synchronize with the selected backplane 8 kHz clock or not.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:INPut[1..n]:TELEcom:BACKplane:ALARm: STATUs?<wsp>LOC</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter is:</p> <p>LOC</p> <p>Sets the status of the backplane 8 kHz clock for synchronization.</p> <p>LOC, selects the LOC (Loss Of Clock) as the alarm status for the input port.</p>

**:INPut[1..n]:TELeom:BACKplane:ALARm:
STATus?**

Response Syntax <Status>

Response(s)

Status:

The response data syntax for <Status> is defined as <CHARACTER RESPONSE DATA> element.

Returns the status of the clock synchronization.

PRESENT, indicates the module is able to synchronize with the selected test clock.

ABSENT, indicates the module is not able to synchronize with the selected test clock.

Example(s)

* INP:TEL:BACK:ALAR:STAT? LOS Returns the status of the clock synchronization.

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

Description	<p>This query returns the generated frequency at the REF OUT port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:OUTPut[1..n]:TELEcom:REFoutput:FREQuency?
Parameter(s)	None
Response Syntax	<Frequency>
Response(s)	<p>Frequency:</p> <p>The response data syntax for <Frequency> is defined as <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the generated frequency.</p>
Example(s)	* OUTP:TEL:REF:FREQ? Returns the generated frequency.

**:OUTPut[1..n]:TELEcom:REFoutput:
SIGNal:STATus?**

Description	<p>This query returns the status of the signal at the REF OUT port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:OUTPut[1..n]:TELEcom:REFoutput:SIGNal:STATus?<wsp>OUTPut</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter is:</p> <p>OUTPut</p> <p>Sets the status of the signal.</p> <p>OUTPut, selects output as the status of the signal.</p>

**:OUTPut[1..n]:TELEcom:REFoutput:
SIGNal:STATus?****Response Syntax** <Status>**Response(s)**

Status:

The response data syntax for <Status> is defined as <CHARACTER RESPONSE DATA> element.

Returns the status of a signal.

PRESENT, indicates the signal is present.

ABSENT, indicates the signal is absent.

Example(s)

* OUTP:TEL:REF:SIGN:STAT? OUTP Returns the status of the signal.

:OUTPut[1..n]:TELecom:REFOutput:DRATio

Description This command sets the selection of the transmit test clock divider ratio.

At *RST, this value is set to C16.

Syntax :OUTPut[1..n]:TELecom:REFOutput:DRATio
<wsp>C16|C32|C64

:OUTPut[1..n]:TELEcom:REFoutput:DRATio**Parameter(s)**

Dratio:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

C16|C32|C64

Sets the ratio for the clock output frequency.

C16, selects C16 as the ratio for the clock output frequency.

C32, selects C32 as the ratio for the clock output frequency.

C64, selects C64 as the ratio for the clock output frequency.

Example(s)

* OUTP:TEL:REF:DRAT C16

* OUTP:TEL:REF:DRAT? Returns C16

See Also

* OUTPut[1..n]:TELEcom:REFoutput:DRATio?

:OUTPut[1..n]:TELEcom:REFOutput:DRATio?

Description This query returns the transmit test clock divider ratio.

At *RST, this value is set to C16.

Syntax :OUTPut[1..n]:TELEcom:REFOutput:DRATio?

Parameter(s) None

Response Syntax <Dratio>

:OUTPut[1..n]:TELEcom:REFoutput:DRATio?

Response(s)	<p>Dratio:</p> <p>The response data syntax for <Dratio> is defined as <CHARACTER RESPONSE DATA> element.</p> <p>Returns the transmit test clock divider ratio.</p> <p>C16, selects C16 as the ratio for the clock output frequency.</p> <p>C32, selects C32 as the ratio for the clock output frequency.</p> <p>C64, selects C64 as the ratio for the clock output frequency.</p>
Example(s)	<ul style="list-style-type: none">* OUTP:TEL:REF:DRAT C16* OUTP:TEL:REF:DRAT? Returns C16
See Also	<ul style="list-style-type: none">* OUTPut[1..n]:TELEcom:REFoutput:DRATio

Port Command Reference

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

Description	<p>This query returns the frequency of signal of the optical port for the transmitter.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQUENCY?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Frequency></p>
Response(s)	<p>Frequency: The response data syntax for <Frequency> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the frequency of signal of the optical port for the transmitter.</p>
Example(s)	<p>* SOUR:DATA:TEL:OPT:PORT:FREQ? Returns the frequency of signal of the optical port.</p>

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

Description	<p>This query returns the nominal frequency of the signal.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQuency:NOMinal?
Parameter(s)	None
Response Syntax	<Nominal>
Response(s)	<p>Nominal:</p> <p>The response data syntax for <Nominal> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the nominal frequency of the signal.</p>
Example(s)	<p>* SOUR:DATA:TEL:OPT:PORT:FREQ:NOM?</p> <p>Returns the nominal frequency of the signal.</p>

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

Description This command sets the offset value between the standard rate specification and the rate of input signal for the transmitter of optical port.

At *RST, this value is set to 0.

Syntax :SOURce[1..n]:DATA:TELEcom:OPTical:PORT:
FREQUency:OFFSet:VALue<wsp><Value>
|MAXimum|MINimum

Parameter(s) Value:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum|MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.

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

Sets the offset value between the standard rate specification and the rate of input signal for the transmitter of optical port.

Choices are -50 through +50.

Example(s)

* SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL 15

* SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL?

Returns the frequency offset value of the input signal.

See Also

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

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

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

Description	<p>This query returns the offset value between the standard rate specification and the rate of input signal of the transmitter of optical port.</p> <p>At *RST, this value is set to 0.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQuency:OFFSet:VALue? [<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the frequency offset value will be returned.</p>
Response Syntax	<p><Value></p>

**:SOURCE[1..n]:DATA:TELEcom:OPTical:PORT:
FREQUENCY:OFFSet:VALue?**

Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the offset value between the standard rate specification and the rate of input signal.</p>
Example(s)	<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>
See Also	<p>* SOURCE[1..n]:DATA:TELEcom:OPTical: FREQUENCY:OFFSet:VALue</p> <p>* SOURCE[1..n]:DATA:TELEcom:OPTical: FREQUENCY:OFFSet</p>

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

Description	<p>This command enables or disables the frequency offset generation of the optical port.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:OFFset <wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the frequency offset generation.</p>
Example(s)	<p>* 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</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OPTical: FREQUency:OFFSet:VALue * SOURce[1..n]:DATA:TELEcom:OPTical: FREQUency:OFFSet?</p>

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

Description	<p>This query returns the frequency offset generation of the optical port.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:OFFSet?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the frequency offset generation of the optical port.</p>
Example(s)	<p>* 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</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OPTical: FREQUency:OFFSet:VALue * SOURce[1..n]:DATA:TELEcom:OPTical: FREQUency:OFFSet</p>

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

Description	<p>This query returns the frequency of signal of the optical port for the receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Frequency></p>
Response(s)	<p>Frequency: The response data syntax for <Frequency> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the frequency of signal of the optical port for the receiver.</p>
Example(s)	<p>* SENS:DATA:TEL:OPT:PORT:FREQ? Returns the frequency of signal of the optical port.</p>

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

Description	<p>This query returns the frequency offset between the standard rate specification and the largest rate recorded from the received signal of optical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:POSitive?
Parameter(s)	None
Response Syntax	<Positive>

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

Response(s)	Positive: The response data syntax for <Positive> is defined as a <STRING RESPONSE DATA> element. Returns the positive frequency.
Example(s)	* SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL 15 * SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS ON * SENS:DATA:TEL:OPT:PORT:FREQ:POS? Returns the positive frequency.
See Also	* SOURce[1..n]:DATA:TELEcom:OPTical:FREQuency:OFFSet:VALue? * SOURce[1..n]:DATA:TELEcom:OPTical:FREQuency:OFFSet

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

Description	<p>This query returns the frequency offset between the standard rate specification and the smallest rate recorded from the received signal of optical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:NEGative?
Parameter(s)	None
Response Syntax	<Negative>

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

Response(s)	Negative: The response data syntax for <Negative> is defined as a <STRING RESPONSE DATA> element. Returns the negative frequency value.
Example(s)	* SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL 15 * SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS ON * SENS:DATA:TEL:OPT:PORT:FREQ:NEG? Returns the negative frequency value.
See Also	* SOURce[1..n]:DATA:TELEcom:OPTical: FREQuency:OFFSet:VALue? * SOURce[1..n]:DATA:TELEcom:OPTical: FREQuency:OFFSet

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

Description	<p>This query returns the frequency offset value between the standard rate specification and the rate of input signal of the receiver of optical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:OPTical:PORT: FREQUency:OFFSet:VALue?
Parameter(s)	None
Response Syntax	<Value>

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

Response(s)	Value: The response data syntax for <Value> is defined as a <STRING RESPONSE DATA> element. Returns the frequency offset value.
Example(s)	* SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL 15 * SOUR:DATA:TEL:OPT:PORT:FREQ:OFFS ON * SENS:DATA:TEL:OPT:PORT:FREQ:OFFS:VAL? Returns the frequency offset value.
See Also	* SOURce[1..n]:DATA:TELEcom:OPTical: FREQuency:OFFSet:VALue * SOURce[1..n]:DATA:TELEcom:OPTical: FREQuency:OFFSet

**:FETCh[1..n]:DATA:TELEcom:OPTical:INPut:
STATUs?**

Description	<p>This query returns the presence of input signal of the optical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OPTical:INPut: STATUs?
Parameter(s)	None
Response Syntax	<Status>
Response(s)	<p>Status:</p> <p>The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> element. Returns the status of the input signal.</p> <p>PRESENT, indicates the presence of input signal at the optical port.</p> <p>ABSENT, indicates the absence of input signal at the optical port.</p>
Example(s)	<p>* FETC:DATA:TEL:OPT:INP:STAT?</p> <p>Returns the status of signal.</p>

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

Description	<p>This query returns the presence of output signal of the optical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OPTical:OUTPut:STATus?
Parameter(s)	None
Response Syntax	<Status>
Response(s)	<p>Status:</p> <p>The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> element. Returns the status of the output signal. PRESENT, indicates the presence of output signal at the optical port. ABSENT, indicates the absence of output signal at the optical port.</p>
Example(s)	<p>* FETC:DATA:TEL:OPT:OUTP:STAT?</p> <p>Returns the status of the output signal.</p>

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

Description	<p>This query returns the receiver power level in decibel milli watts.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:OPTical:PORT: PLEVel?
Parameter(s)	None
Response Syntax	<Plevel>
Response(s)	<p>Plevel:</p> <p>The response data syntax for <Plevel> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the receiver power level.</p>
Example(s)	<p>* SENS:DATA:TEL:OPT:PORT:PLEV?</p> <p>Returns the receiver power level.</p>

**:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:
PORT:TYPE**

Description	<p>This command sets the type of optical port alarm.</p> <p>At *RST, this value is set to LOS.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:TYPE<wsp>LOS</p>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> element for this parameter is: LOS. Selects the type of port alarm for the instrument. LOS, selects the LOS (Loss of Signal) as port alarm type.</p>
Example(s)	<p>* SOUR[1..n]:DATA:TEL:OPT:ALAR:PORT:TYPE LOS * SOUR[1..n]:DATA:TEL:OPT:ALAR:PORT:TYPE? Returns LOS</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OPTical:ALARm :PORT:TYPE?</p>

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

Description	This query returns the type of optical port alarm. At *RST, this value is set to LOS.
Syntax	:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	Alarm: The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of port alarm. LOS, Loss of Signal (LOS) is selected as port alarm.
Example(s)	* SOUR[1..n]:DATA:TEL:OPT:ALAR:PORT:TYPE LOS * SOUR[1..n]:DATA:TEL:OPT:ALAR:PORT:TYPE? Returns LOS
See Also	* SOURce[1..n]:DATA:TELEcom:OPTical:ALARm :PORT:TYPE

:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm: PORT

Description	<p>This command enables or disables the optical port alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm: PORT <wsp> <Set></p>
Parameter(s)	<p>Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0.</p>

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

Enables or disables the port alarm generation.

Example(s)

* SOUR:DATA:TEL:OPT:PORT:ALAR:PORT:TYPE
LOS
* SOUR:DATA:TEL:OPT:PORT:ALAR:PORT ON
* SOUR:DATA:TEL:OPT:PORT:ALAR:PORT?
Returns 1

See Also

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

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

Description	<p>This query returns the status of optical port alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OPTical:ALARm: PORT?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of port alarm generation.</p>
Example(s)	<p>* SOUR:DATA:TEL:OPT:ALAR:PORT:TYPE LOS * SOUR:DATA:TEL:OPT:ALAR:PORT ON * SOUR:DATA:TEL:OPT:ALAR:PORT? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OPTical:ALARm :PORT:TYPE * SOURce[1..n]:DATA:TELEcom:OPTical:ALARm :PORT</p>

**:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:
PORT:HISTory?**

Description	<p>This query returns the history status of optical port alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:HISTory?<wsp>LOS FREQuency</pre>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOS FREQuency. Selects the type of port alarm. LOS, selects the type of port alarm as LOS (Loss of Signal). FREQuency, selects the type of port alarm as frequency.</p>
Response Syntax	<pre><History></pre>

:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:HISTory?

Response(s)

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the history status of port 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)

* SOUR:DATA:TEL:OPT:ALAR:PORT:TYPE LOS

* SOUR:DATA:TEL:OPT:ALAR:PORT ON

* FETC:DATA:TEL:OPT:ALAR:PORT:HIST? LOS

Returns the alarm history.

See Also

* SOURce[1..n]:DATA:TELEcom:OPTical:ALARm
:PORT:TYPE

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

**:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:
PORT:CURREnt?**

Description	<p>This query returns the current status of optical port alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:CURREnt?<wsp>LOS FREQUENCY</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>LOS FREQUENCY.</p> <p>Selects the type of port alarm.</p> <p>LOS, selects the type of port alarm as LOS (Loss of Signal).</p> <p>FREQUENCY, selects the type of port alarm as frequency.</p>
Response Syntax	<pre><Current></pre>

:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:CURREnt?

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current status of port 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.

Example(s)

* SOUR:DATA:TEL:OPT:ALAR:PORT:TYPE LOS
* SOUR:DATA:TEL:OPT:ALAR:PORT ON
* FETC:DATA:TEL:OPT:ALAR:PORT:CURR? LOS
Returns the current alarm status.

See Also

* SOURce[1..n]:DATA:TELEcom:OPTical:ALARm
:PORT:TYPE
* SOURce[1..n]:DATA:TELEcom:OPTical:ALARm
:PORT

**:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm:
PORT:SEConds?**

Description	<p>This query returns the number of seconds within which optical port alarm occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:SEConds?<wsp>LOS FREQUENCY</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>LOS FREQUENCY</p> <p>Selects the type of port alarm.</p> <p>LOS, selects the type of port alarm as LOS (Loss of Signal).</p> <p>FREQUENCY, selects the type of port alarm as frequency.</p>
Response Syntax	<pre><Seconds></pre>

:FETCh[1..n]:DATA:TELEcom:OPTical:ALARm: PORT:SEConds?

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of port alarm.
Example(s)	* SOUR:DATA:TEL:OPT:ALAR:PORT:TYPE LOS * SOUR:DATA:TEL:OPT:ALAR:PORT ON * FETC:DATA:TEL:OPT:ALAR:PORT:SEC? LOS Returns the number of seconds of port alarm.
See Also	* SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT:TYPE * SOURce[1..n]:DATA:TELEcom:OPTical:ALARm:PORT

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

Description	<p>This command enables or disables the state of laser.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:OUTPut[1..n]:TELEcom:LASer<wsp><Set>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the laser.</p>
Example(s)	<p>* OUTP:TEL:LAS ON</p> <p>* OUTP:TEL:LAS? Returns 1</p>
See Also	* OUTPut[1..n]:TELEcom:LASer?

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

Description	This query returns the current state of laser. At *RST, this value is set to OFF.
Syntax	:OUTPut[1..n]:TELEcom:LASer?
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the state of laser.
Example(s)	* OUTP:TEL:LAS ON * OUTP:TEL:LAS? Returns 1
See Also	* OUTPut[1..n]:TELEcom:LASer

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

Description	<p>This query returns the wavelength of the optical output port in discrete form.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:OUTPut[1..n]:TELEcom:LASer:WAVelength?
Parameter(s)	None
Response Syntax	<Wavelength>
Response(s)	<p>Wavelength:</p> <p>The response data syntax for <Wavelength> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the wavelength.</p> <p>NM1310, 1310 nm wavelength is selected.</p> <p>NM1550, 1550 nm wavelength is selected.</p> <p>UNKNOWN, wavelength is Unknown.</p>
Example(s)	<p>* OUTP:TEL:LAS ON</p> <p>* OUTP:TEL:LAS:WAV?</p> <p>Returns the wavelength of the optical output port</p>
See Also	* OUTPut[1..n]:TELEcom:LASer

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
OUTPut:STATus?**

Description	<p>This query returns the presence of output signal of the electrical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ELECtrical:OUTPut:STATus?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Status></p>
Response(s)	<p>Status: The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> element. Sets the status of output signal. PRESENT, indicates the presence of output signal at the electrical port. ABSENT, indicates the absence of output signal at the electrical port.</p>
Example(s)	<p>* FETC:DATA:TEL:ELEC:OUTP:STAT? Returns the status of output signal.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
INPut:STATus?**

Description	<p>This query returns the presence of input signal of the electrical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:ELECtrical:INPut:STATus?
Parameter(s)	None
Response Syntax	<Status>
Response(s)	<p>Status:</p> <p>The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> element. Returns the status of input signal.</p> <p>PRESENT, indicates the presence of input signal at the electrical port.</p> <p>ABSENT, indicates the absence of input signal at the electrical port.</p>
Example(s)	<p>* FETC:DATA:TEL:ELEC:INP:STAT?</p> <p>Returns the status of the input signal.</p>
Note	FTB/IQS-8140 Transport Blazer does not support this query.

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

Description	<p>This query returns the frequency of signal of the electrical port for the transmitter.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ELECtrical:PORT: FREQuency?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Frequency></p>
Response(s)	<p>Frequency: The response data syntax for <Frequency> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the frequency of signal of the electrical port for the transmitter.</p>
Example(s)	<p>* SOUR:DATA:TEL:ELEC:PORT:FREQ? Returns the frequency of signal of the electrical port.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>

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

Description	<p>This query returns the nominal frequency of the signal.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:ELECTrical:PORT: FREQUency:NOMinal?
Parameter(s)	None
Response Syntax	<Nominal>
Response(s)	<p>Nominal:</p> <p>The response data syntax for <Nominal> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the nominal frequency of the signal.</p>
Example(s)	<p>* SOUR:DATA:TEL:ELEC:PORT:FREQ:NOM?</p> <p>Returns the nominal frequency of the signal.</p>
Note	FTB/IQS-8140 Transport Blazer does not support this query.

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

Description This command sets the frequency offset value between the standard rate specification and the rate of input signal for the transmitter of electrical port.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:ELECTrical:PORT:
FREQUency:OFFSet:VALue <wsp> <Value>
|MAXimum|MINimum

Parameter(s) Value:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum|MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the offset value of electrical port.

**:SOURce[1..n]:DATA:TELecom:ELECtrical:PORT:
FREQuency:OFFSet:VALue**

Example(s)	* SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS:VAL 15 * SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS:VAL? Returns 15
See Also	* SOURce[1..n]:DATA:TELecom:ELECtrical: FREQuency:OFFSet:VALue? * SOURce[1..n]:DATA:TELecom:ELECtrical: FREQuency:OFFSet
Note	FTB/IQS-8140 Transport Blazer does not support this command.

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

Description This query returns the frequency offset value between the standard rate specification and the rate of input signal for transmitter of electrical port.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:OPTIcal:PORT:
FREQuency:OFFSet:VALue? [<wsp>MAXimum |
MINimum]

Parameter(s) The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum | MINimum.
MAXimum is used to retrieve the instrument's greatest supported value.
MINimum is used to retrieve the instrument's smallest supported value.
This parameter is optional. If no token is specified, the current frequency offset value will be returned.

Response Syntax <Value>

**:SOURce[1..n]:DATA:TELecom:ELECtrical:PORT:
FREQuency:OFFSet:VALue?**

Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the offset value between the standard rate specification and the rate of input signal</p>
Example(s)	<p>* SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS:VAL 15</p> <p>* SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS:VAL? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELecom:ELECtrical: FREQuency:OFFSet:VALue</p> <p>* SOURce[1..n]:DATA:TELecom:ELECtrical: FREQuency:OFFSet</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>

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

Description	<p>This command enables or disables the frequency offset generation of electrical port.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ELECtrical:PORT: FREQuency:OFFSet<wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the frequency offset generation.</p>

**:SOURce[1..n]:DATA:TELecom:ELECtrical:PORT:
FREQUency:OFFSet**

Example(s)	* SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS:VAL 15 * SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS ON * SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS Returns 1
See Also	* SOURce[1..n]:DATA:TELecom:ELECtrical: FREQUency:OFFSet:VALue * SOURce[1..n]:DATA:TELecom:ELECtrical: FREQUency:OFFSet?
Note	FTB/IQS-8140 Transport Blazer does not support this command.

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

Description	This query returns the frequency offset generation of electrical port. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:ELECTrical:PORT: FREQuency:OFFSet?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the frequency offset generation of electrical port.

**:SOURce[1..n]:DATA:TELecom:ELECtrical:PORT:
FREQuency:OFFSet?**

Example(s)	* SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS:VAL 15 * SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS ON * SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELecom:ELECtrical: FREQuency:OFFSet:VALue * SOURce[1..n]:DATA:TELecom:ELECtrical: FREQuency:OFFSet

**:SENSe[1..n]:DATA:TELEcom:ELECTrical:PORT:
PLEVel?**

Description	<p>This query returns the electrical port receiver power level in decibel milli watts.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:ELECTrical:PORT: PLEVel?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Plevel></p>
Response(s)	<p>Plevel: The response data syntax for <Plevel> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the receiver power level.</p>
Example(s)	<p>* SENS:DATA:TEL:ELEC:PORT:PLEV? Returns the receiver power level.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>

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

Description	<p>This query returns the frequency of signal of the electrical port for the receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:SENSe[1..n]:DATA:TELEcom:ELECtrical:PORT: FREQUency?</code>
Parameter(s)	None
Response Syntax	<code><Frequency></code>
Response(s)	<p>Frequency:</p> <p>The response data syntax for <code><Frequency></code> is defined as a <code><NR2 NUMERIC RESPONSE DATA></code> element.</p> <p>Returns the frequency of signal of the electrical port for the receiver.</p>
Example(s)	<p>* SENS:DATA:TEL:ELEC:PORT:FREQ?</p> <p>Returns the frequency of signal of the electrical port.</p>
Note	FTB/IQS-8140 Transport Blazer does not support this query.

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

Description	<p>This query returns the offset between the standard rate specification and the smallest rate recorded from the received signal of electrical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:ELECtrical:PORT: FREQuency:NEGative?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Negative></p>
Response(s)	<p>Negative: The response data syntax for <Negative> is defined as a <STRING RESPONSE DATA> element. Returns the negative frequency of the input signal.</p>

**:SENSe[1..n]:DATA:TELeom:ELECtrical:PORT:
FREQUency:NEGative?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS:VAL 15* SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS ON* SENS:DATA:TEL:ELEC:PORT:FREQ:NEG? Returns the negative frequency.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELeom:OPTical: FREQUency:OFFSet:VALue?* SOURce[1..n]:DATA:TELeom:OPTical: FREQUency:OFFSet

**:SENSe[1..n]:DATA:TELEcom:ELECTrical:PORT:
FREQuency:OFFSet:Value?**

Description	<p>This query returns the offset value between the standard rate specification and the rate of input signal for receiver of electrical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:ELECTrical:PORT: FREQuency:OFFSet:VALue?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Offset></p>
Response(s)	<p>Offset: The response data syntax for <Offset> is defined as a <STRING RESPONSE DATA> element. Returns the offset value between the standard rate specification and the rate of input signal.</p>

**:SENSe[1..n]:DATA:TELecom:ELECTrical:PORT:
FREQUency:OFFSet:Value?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS:VAL 15* SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS ON* SENS:DATA:TEL:ELEC:PORT:FREQ:OFFS:VAL? Returns the frequency offset value.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELecom:OPTical: FREQUency:OFFSet:VALue?* SOURce[1..n]:DATA:TELecom:OPTical: FREQUency:OFFSet

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

Description	<p>This query returns the offset between the standard rate specification and the largest rate recorded from the received signal of electrical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:ELECtrical:PORT: FREQuency:POSitive?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Positive></p>
Response(s)	<p>Positive: The response data syntax for <Positive> is defined as a <STRING RESPONSE DATA> element. Returns the positive frequency of the input signal.</p>

**:SENSe[1..n]:DATA:TELecom:ELECtrical:PORT:
FREQuency:POSitive?**

Example(s)	* SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS:VAL 15 * SOUR:DATA:TEL:ELEC:PORT:FREQ:OFFS ON * SENS:DATA:TEL:ELEC:PORT:FREQ:POS? Returns the positive frequency.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELecom:OPTical: FREQuency:OFFSet:VALue? * SOURce[1..n]:DATA:TELecom:OPTical: FREQuency:OFFSet

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:PORT:
AMPLitude?**

Description	<p>This query returns the amplitude of the electrical port.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ELECtrical:PORT: AMPLitude?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Amplitude></p>
Response(s)	<p>Amplitude: The response data syntax for <Amplitude> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the amplitude of the electrical port.</p>
Example(s)	<p>* FETC:DATA:TEL:ELEC:PORT:AMPL? Returns the amplitude of the electrical port.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT:TYPE**

Description	<p>This command sets the type of electrical port alarm.</p> <p>At *RST, this value is set to LOS.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:TYPE<wsp>LOS</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: LOS.</p> <p>Selects the type of port alarm for the instrument. LOS, selects the LOS (Loss of Signal) as port alarm type.</p>

:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:TYPE

Example(s)	* SOUR:DATA:TEL:ELEC:ALAR:PORT:TYPE LOS * SOUR:DATA:TEL:ELEC:ALAR:PORT:TYPE? Returns LOS
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:TYPE?

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT:TYPE?**

Description	<p>This query returns the type of electrical port alarm.</p> <p>At *RST, this value is set to LOS.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of port alarm.</p> <p>LOS, Loss of Signal (LOS) is selected as port alarm.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT:TYPE?**

Example(s)	* SOUR:DATA:TEL:ELEC:ALAR:PORT:TYPE LOS * SOUR:DATA:TEL:ELEC:ALAR:PORT:TYPE? Returns LOS
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:TYPE

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT**

Description	<p>This command enables or disables the electrical port alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT <wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the port alarm generation.</p>

:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT

Example(s)

- * SOUR:DATA:TEL:ELEC:ALAR:PORT:TYPE LOS
- * SOUR:DATA:TEL:ELEC:ALAR:PORT ON
- * SOUR:DATA:TEL:ELEC:ALAR:PORT? Returns 1

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT:TYPE
- * SOURce[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT:TYPE?
- * SOURce[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT?

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT?**

Description	<p>This query returns the status of port alarm occurred.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT?</code>
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of port alarm generation.</p>

:SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT?

Example(s)

- * SOUR:DATA:TEL:ELEC:ALAR:PORT:TYPE LOS
- * SOUR:DATA:TEL:ELEC:ALAR:PORT ON
- * SOUR:DATA:TEL:ELEC:ALAR:PORT? Returns 1

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:ELECtrical:ALARm:PORT:TYPE
- * SOURce[1..n]:DATA:TELEcom:ELECtrical:ALARm:PORT:TYPE?
- * SOURce[1..n]:DATA:TELEcom:ELECtrical:ALARm:PORT

**:SOURce[1..n]:DATA:TELEcom:ELECTrical:
ERRor:PORT:MANual:TYPE**

Description	<p>This command sets the manual type of electrical port error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ELECTrical: ERRor:PORT:MANual:TYPE<wsp>CV BPV EXZ</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CV BPV EXZ.</p> <p>Selects the type of port error.</p> <p>CV, selects the type of port error as CV (Code Violation) for E1, E2, E3, E4, STS-1e/STM-0e, and STS-3e/STM-1e.</p> <p>BPV, selects the type of port error as BPV (Bipolar Violation) for DS1 & DS3.</p> <p>EXZ, selects the type of port error as EXZ (Excessive Zeros) for DS1 & DS3.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:MANual:TYPE**

Example(s)	* SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE CV * SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE? Returns CV
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:MANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:MANual:TYPE?**

Description	<p>This query returns the manual type of electrical port error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:MANual:TYPE?</code>
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of port error.</p> <p>CV, Code Violation (CV) is selected as port error.</p> <p>BPV, Bipolar Violation (BPV) is selected as port error.</p> <p>EXZ, Excessive Zeros (EXZ) is selected as port error.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:MANual:TYPE?**

Example(s)	* SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE CV * SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE? Returns CV
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AMOUNT**

Description	<p>This command sets the amount of electrical port error to be injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AMOUNT <wsp> <Amount> MAXimum MINimum</pre>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the amount of error. Choices are 1 through 50.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AMOut**

Example(s) * SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE
CV
* SOUR:DATA:TEL:ELEC:ERR:PORT:AMO 15
* SOUR:DATA:TEL:ELEC:ERR:PORT:AMO?
Returns 15

Note FTB/IQS-8140 Transport Blazer does not support
this command.

See Also * SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AMOut?

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AMOUnt?**

Description	<p>This query returns the amount of electrical port error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AMOUnt?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<pre><Amount></pre>

:SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AMOut?

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of port error.
Example(s)	* SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE CV * SOUR:DATA:TEL:ELEC:ERR:PORT:AMO 15 * SOUR:DATA:TEL:ELEC:ERR:PORT:AMO? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AMOut

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:INJect**

Description	<p>This command injects the electrical port error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE CV * SOUR:DATA:TEL:ELEC:ERR:PORT:AMO 15 * SOUR:DATA:TEL:ELEC:ERR:PORT:INJ
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AMOunt

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:TYPE**

Description	<p>This command selects the type of Port error for automated injection.</p> <p>At *RST, this value device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor :PORT:AUTomated:TYPE<wsp>CV BPV EXZ</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CV BPV EXZ.</p> <p>Selects the type of Port error for automated injection.</p> <p>CV, selects the type of port error as CV (Code Violation) for E1, E2, E3, E4, STS-1e/STM-0e, and STS-3e/STM-1e.</p> <p>BPV, selects the type of port error as BPV (Bipolar Violation) for DS1 & DS3.</p> <p>EXZ, selects the type of port error as EXZ (Excessive Zeros) for DS1 & DS3.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:TYPE**

Example(s)	* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:TYPE BPV * SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:TYPE? Returns BPV
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:TYPE? * SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:TYPE?**

Description	<p>This query returns the type of Port error for automated injection.</p> <p>At *RST, this value device dependent.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor: PORT:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of Port error for the automated injection.</p> <p>CV, Code Violation (CV) is selected as port error.</p> <p>BPV, Bipolar Violation (BPV) is selected as port error.</p> <p>EXZ, Excessive Zeros (EXZ) is selected as port error.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:TYPE?**

Example(s)	* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:TYPE BPV * SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:TYPE? Returns BPV
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:RATE**

Description	<p>This command sets the injection rate for the selected Port error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor :PORT:AUTomated:RATE<wsp><Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected Port error.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:RATE**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:TYPE BPV* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:RATE 1.0E-10* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:RATE? Returns 1.0E-10
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:RATE?* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Port error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor: PORT:AUTomated:RATE?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injection rate will be returned.</p>
Response Syntax	<p><Rate></p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Port error.</p>
Example(s)	<p>* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:TYPE BPV</p> <p>* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:RATE 1.0E-10</p> <p>* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:RATE? Returns 1.0E-10</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated**

Description	<p>This command enables or disables the selected automated Port error at the rate specified or continuously.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor :PORT:AUTomated <wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated Port error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated****Example(s)**

* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:TYPE
BPV
* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT ON
* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT?
Returns 1

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor: PORT:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated?**

Description	This query returns the status of automated Port error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor: PORT:AUTomated?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated Port error injection.

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated?****Example(s)**

* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:TYPE
BPV
* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT ON
* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT?
Returns 1

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated

:SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:CONTInuous

Description	<p>This command enables or disables the continuous rate of automated Port error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor :PORT:AUTomated:CONTInuous<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated Port error injection continuously.</p>

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:CONTInuous**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:TYPE BPV* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:CONT ON* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:CONT? Returns 1* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT ON
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:CONTInuous?**

Description	This query returns the status of continuous rate of automated Port error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor: PORT:AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuous rate of automated Port error injection.

**:SOURce[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:AUTomated:CONTInuous?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:TYPE BPV* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:CONT ON* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT:CONT? Returns 1* SOUR:DATA:TEL:ELEC:ERR:PORT:AUT ON
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AUTomated:CONTInuous?

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT:HISTory?**

Description	<p>This query returns the history status of electrical port alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:ELECtrical:ALARm:PORT:HISTory?<wsp>LOS FREQuency</code>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>LOS FREQuency.</p> <p>Selects the type of port alarm.</p> <p>LOS, selects the type of port alarm as LOS (Loss of Signal).</p> <p>FREQuency, selects the type of port alarm as frequency.</p>
Response Syntax	<code><History></code>

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ELEC:ALAR:PORT:TYPE LOS* SOUR:DATA:TEL:ELEC:ALAR:PORT ON* FETC:DATA:TEL:ELEC:ALAR:PORT:HIST? LOS <p>Returns the alarm history.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ELECtrical:ALARm:PORT:TYPE* SOURce[1..n]:DATA:TELEcom:ELECtrical:ALARm:PORT

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT:CURREnt?**

Description	<p>This query returns the current status of electrical port alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ELECtrical:ALARm: PORT:CURREnt?<wsp>LOS FREQuency</p>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOS FREQuency. Selects the type of port alarm. LOS, selects the type of port alarm as LOS (Loss of Signal). FREQuency, selects the type of port alarm as frequency.</p>
Response Syntax	<p><Current></p>

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT:CURREnt?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> 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 there is no alarm.</p> <p>INACTIVE, indicates that the test is not running.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ELEC:ALAR:PORT:TYPE LOS* SOUR:DATA:TEL:ELEC:ALAR:PORT ON* FETC:DATA:TEL:ELEC:ALAR:PORT:CURR? LOS <p>Returns the current alarm status.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ELECtrical:ALARm:PORT:TYPE* SOURce[1..n]:DATA:TELEcom:ELECtrical:ALARm:PORT

**:FETCh[1..n]:DATA:TELEcom:ELECTrical:
ALARm:PORT:SEConds?**

Description	<p>This query returns the number of seconds within which electrical port alarm occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ELECTrical:ALARm:PORT:SEConds?<wsp>LOS FREQuency</p>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOS FREQuency Selects the type of port alarm. LOS, selects the type of port alarm as LOS (Loss of Signal). FREQuency, selects the type of port alarm as frequency.</p>
Response Syntax	<p><Seconds></p>

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ALARm:PORT:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of port alarm.
Example(s)	* SOUR:DATA:TEL:ELEC:ALAR:PORT:TYPE LOS * SOUR:DATA:TEL:ELEC:ALAR:PORT ON * FETC:DATA:TEL:ELEC:ALAR:PORT:SEC? LOS Returns the number of seconds of port alarm.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT:TYPE * SOURce[1..n]:DATA:TELEcom:ELECtrical: ALARm:PORT

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:HISTory?**

Description This query returns the history status of electrical port error.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:ELECtrical:ERRor:
PORT:HISTory?<wsp>CV|BPV|EXZ

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
CV|BPV|EXZ.
Selects the type of port error.
CV, selects the type of port error as CV (Code Violation) for E1, E2, E3, E4, STS-1e/STM-0e, and STS-3e/STM-1e.
BPV, selects the type of port error as BPV (Bipolar Violation) for DS1 & DS3.
EXZ, selects the type of port error as EXZ (Excessive Zeros) for DS1 & DS3.

Response Syntax <History>

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of port 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE CV* SOUR:DATA:TEL:ELEC:ERR:PORT:AMO 15* SOUR:DATA:TEL:ELEC:ERR:PORT:INJ* FETC:DATA:TEL:ELEC:ERR:PORT:HIST? CV <p>Returns the current error history.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor:PORT:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor:PORT:AMOUNT* SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor:PORT:INJect

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:SEConds?**

Description	<p>This query returns the number of seconds within which electrical port error occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:ELECtrical:ERRor: PORT:SEConds?<wsp>CV BPV EXZ</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CV BPV EXZ.</p> <p>Selects the type of port error.</p> <p>CV, selects the type of port error as CV (Code Violation) for E1, E2, E3, E4, STS-1e/STM-0e, and STS-3e/STM-1e.</p> <p>BPV, selects the type of port error as BPV (Bipolar Violation) for DS1 & DS3.</p> <p>EXZ, selects the type of port error as EXZ (Excessive Zeros) for DS1 & DS3.</p>
Response Syntax	<code><Seconds></code>

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of port error.
Example(s)	* SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE CV * SOUR:DATA:TEL:ELEC:ERR:PORT:AMO 15 * SOUR:DATA:TEL:ELEC:ERR:PORT:INJ * FETC:DATA:TEL:ELEC:ERR:PORT:SEC? CV Returns the number of errored seconds.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AMOUNT * SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:INJect

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:CURRent?**

Description	<p>This query returns the current status of electrical port error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:ELECtrical:ERRor: PORT:CURRent?<wsp>CV BPV EXZ</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CV BPV EXZ.</p> <p>Selects the type of port error.</p> <p>CV, selects the type of port error as CV (Code Violation) for E1, E2, E3, E4, STS-1e/STM-0e, and STS-3e/STM-1e.</p> <p>BPV, selects the type of port error as BPV (Bipolar Violation) for DS1 & DS3.</p> <p>EXZ, selects the type of port error as EXZ (Excessive Zeros) for DS1 & DS3.</p>
Response Syntax	<code><Current></code>

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:
ERRor:PORT:CURREnt?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of port 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE CV* SOUR:DATA:TEL:ELEC:ERR:PORT:AMO 15* SOUR:DATA:TEL:ELEC:ERR:PORT:INJ* FETC:DATA:TEL:ELEC:ERR:PORT:CURR? CV <p>Returns the current error status.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor:PORT:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor:PORT:AMOUNT* SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor:PORT:INJect

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:ERROr:
PORT:COUNT?**

Description	<p>This query returns the count of electrical port error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ELECtrical:ERROr: PORT:COUNT? <wsp>CV BPV EXZ</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CV BPV EXZ.</p> <p>Selects the type of port error.</p> <p>CV, selects the type of port error as CV (Code Violation) for E1, E2, E3, E4, STS-1e/STM-0e, and STS-3e/STM-1e.</p> <p>BPV, selects the type of port error as BPV (Bipolar Violation) for DS1 & DS3.</p> <p>EXZ, selects the type of port error as EXZ (Excessive Zeros) for DS1 & DS3.</p>
Response Syntax	<p><Count></p>

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:ERRor:
PORT:COUNT?**

Response(s)	Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of port error.
Example(s)	* SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE CV * SOUR:DATA:TEL:ELEC:ERR:PORT:AMO 15 * SOUR:DATA:TEL:ELEC:ERR:PORT:INJ * FETC:DATA:TEL:ELEC:ERR:PORT:COUN? CV Returns the error count.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:AMOUNT * SOURce[1..n]:DATA:TELEcom:ELECtrical: ERRor:PORT:INJect

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:ERROr:
PORT:RATE?**

Description	<p>This query returns the current rate of port error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ELECtrical:ERROr: PORT:RATE?<wsp>CV BPV EXZ</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CV BPV EXZ.</p> <p>Selects the type of port error.</p> <p>CV, selects the type of port error as CV (Code Violation) for E1, E2, E3, E4, STS-1e/STM-0e, and STS-3e/STM-1e.</p> <p>BPV, selects the type of port error as BPV (Bipolar Violation) for DS1 & DS3.</p> <p>EXZ, selects the type of port error as EXZ (Excessive Zeros) for DS1 & DS3.</p>
Response Syntax	<p><Rate></p>

**:FETCh[1..n]:DATA:TELEcom:ELECtrical:ERRor:
PORT:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of port error.</p>
Example(s)	<p>* SOUR:DATA:TEL:ELEC:ERR:PORT:MAN:TYPE CV</p> <p>* SOUR:DATA:TEL:ELEC:ERR:PORT:AMO 15</p> <p>* SOUR:DATA:TEL:ELEC:ERR:PORT:INJ</p> <p>* FETC:DATA:TEL:ELEC:ERR:PORT:RATE? CV</p> <p>Returns the current error rate.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor:PORT:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor:PORT:AMOUNT</p> <p>* SOURce[1..n]:DATA:TELEcom:ELECtrical:ERRor:PORT:INJect</p>

Pattern Command Reference

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

Description

This command selects the type of payload pattern for the transmitter.

At *RST, this value is device dependent.

Syntax

:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE
<wsp>PRBS2E9|PRBS2E11|PRBS2E15|
PRBS2E20|PRBS2E23|PRBS2E31|P1100|P1010|
P1111|P0000|P1IN8|P1IN16|P3IN24|QRSS|
DALY|P55OCTET|NCLient|UPATtern

Parameter(s)

Pattern:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
PRBS2E9|PRBS2E11|PRBS2E15|PRBS2E20|
PRBS2E23|PRBS2E31|P1100|P1010|
P1111|P0000|P1IN8|P1IN16|P3IN24|QRSS|
DALY|P55OCTET|NCLient|UPATtern.
Selects the type of payload pattern.
PRBS2E9, selects the pattern type as PRBS (Pseudo Random Bit Sequence) 2^9-1 .
PRBS2E11, selects the pattern type as PRBS $2^{11}-1$.

:SOURCE[1..n]:DATA:TELEcom:PATtern:TYPE

PRBS2E15, selects the pattern type as PRBS $2^{15}-1$.

PRBS2E20, selects the pattern type as PRBS $2^{20}-1$.

PRBS2E23, selects the pattern type as PRBS $2^{23}-1$.

PRBS2E31, selects the pattern type as PRBS $2^{31}-1$.

P1100, selects the pattern type as 1100.

P1010, selects the pattern type as 1010.

P1111, selects the pattern type as 1111.

P0000, selects the pattern type as 0000.

P1IN8, selects the pattern type as 1 in 8.

P1IN16, selects the pattern type as 1 in 16.

P3IN24, selects the pattern type as 3 in 24.

QRSS, selects the pattern type as QRSS (Quasi-Random Signal Source) for DS1/DS3.

DALY, selects the pattern type as DALY for DS1/DS3.

P55OCTET, selects the pattern type as 55OCTET for DS1/DS3.

NCLient, selects the pattern type as Null Client for DS1/DS3.

UPATtern, selects the pattern type as User Pattern.

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

Example(s) * SOUR:DATA:TEL:PATT:TYPE PRBS2E9
 * SOUR:DATA:TEL:PATT:TYPE? Returns PRBS2E9

Note For **8120NGE/8130NGE/8130NGEv2** modules, choices are PRBS2E9|PRBS2E11|PRBS2E15|PRBS2E20|PRBS2E23|PRBS2E31|P1100|P1010|P1111|P0000|P1IN8|P1IN16|P3IN24|QRSS|DALY|P55OCTET|NCLient|UPATtern.
 For **8140 module**, choices are PRBS2E9|PRBS2E11|PRBS2E15|PRBS2E20|PRBS2E23|PRBS2E31|P1100|P1010|P1111|P0000|P1IN8|P1IN16|NCLient|UPATtern.

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

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

Description	This query returns the type of payload pattern for the transmitter. At *RST, this value is device dependent.
Syntax	:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE?
Parameter(s)	None
Response Syntax	<Pattern>
Response(s)	<p>Pattern:</p> <p>The response data syntax for <Pattern> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of payload pattern.</p> <p>PRBS2E9, pattern type Pseudo Random Bit Sequence (PRBS) 2^9-1 is selected.</p> <p>PRBS2E11, pattern type PRBS $2^{11}-1$ is selected.</p> <p>PRBS2E15, pattern type PRBS $2^{15}-1$ is selected.</p> <p>PRBS2E20, pattern type PRBS $2^{20}-1$ is selected.</p> <p>PRBS2E23, pattern type PRBS $2^{23}-1$ is selected.</p> <p>PRBS2E31, pattern type PRBS $2^{31}-1$ is selected.</p> <p>P1100, pattern type 1100 is selected.</p> <p>P1010, pattern type 1010 is selected.</p>

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

P1111, pattern type 1111 is selected.
P0000, pattern type 0000 is selected.
P1IN8, 1 in 8 as pattern type is selected.
P1IN16, 1 in 16 as pattern type is selected.
P3IN24, 3 in 24 as pattern type is selected.
QRSS, Quasi-Random Signal Source (QRSS) pattern type is selected.
DALY, DALY pattern type is selected.
P55OCTET, 55OCTET pattern type is selected.
NCLIENT, Null Client pattern type is selected.
UPATTERN, User Pattern is selected.

Example(s)

* SOUR:DATA:TEL:PATT:TYPE PRBS2E9
* SOUR:DATA:TEL:PATT:TYPE? Returns PRBS2E9

See Also

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

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

This command selects the type of payload pattern for the receiver.

At *RST, this value is device dependent.

Syntax

```
:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE
<wsp>PRBS2E9|PRBS2E11|PRBS2E15|
PRBS2E20|PRBS2E23|PRBS2E31|P1100|P1010|
P1111|P0000|P1IN8|P1IN16|P3IN24|QRSS|
DALY|P55OCTET|NCLient|UPATtern.
```

Parameter(s)

Pattern:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

```
PRBS2E9|PRBS2E11|PRBS2E15|PRBS2E20|
PRBS2E23|PRBS2E31|UPATtern|P1100|P1010|
P1111|P0000|P1IN8|P1IN16|P3IN24|QRSS|
DALY|P55OCTET|NCLient.
```

Selects the test pattern.

PRBS2E9, selects the pattern type as PRBS (Pseudo Random Bit Sequence) 2^9-1 .

PRBS2E11, selects the pattern type as PRBS $2^{11}-1$.

PRBS2E15, selects the pattern type as PRBS $2^{15}-1$.

PRBS2E20, selects the pattern type as PRBS $2^{20}-1$.

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

PRBS2E23, selects the pattern type as PRBS $2^{23}-1$.

PRBS2E31, selects the pattern type as PRBS $2^{31}-1$.

P1100, selects the pattern type as 1100.

P1010, selects the pattern type as 1010.

P1111, selects the pattern type as 1111.

P0000, selects the pattern type as 0000.

P1IN8, selects the pattern type as 1 in 8.

P1IN16, selects the pattern type as 1 in 16.

P3IN24, selects the pattern type as 3 in 24.

QRSS, selects the pattern type as QRSS (Quasi-Random Signal Source) for DS1/DS3.

DALY, selects the pattern type as DALY for DS1/DS3.

P55OCTET, selects the pattern type as 55OCTET for DS1/DS3.

NCLient, selects the pattern type as Null Client for DS1/DS3.

UPATtern, selects the pattern type as User Pattern.

:SENSe[1..n]:DATA:TELecom:PATtern:TYPE

Example(s) * SENS:DATA:TEL:PATT:TYPE PRBS2E9
 * SENS:DATA:TEL:PATT:TYPE? Returns PRBS2E9

Note For **8120NGE/8130NGE/8130NGEv2** modules, choices are PRBS2E9|PRBS2E11|PRBS2E15|PRBS2E20|PRBS2E23|PRBS2E31|P1100|P1010|P1111|P0000|P1IN8|P1IN16|P3IN24|QRSS|DALY|P55OCTET|NCLient|UPATtern.
 For **8140 module**, choices are PRBS2E9|PRBS2E11|PRBS2E15|PRBS2E20|PRBS2E23|PRBS2E31|P1100|P1010|P1111|P0000|P1IN8|P1IN16|NCLient|UPATtern.

See Also * SENSe[1..n]:DATA:TELecom:PATtern:TYPE?

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

Description This query returns the type of payload pattern for the receiver.

At *RST, this value is device dependent.

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

Parameter(s) None

Response Syntax <Pattern>

Response(s) Pattern:
The response data syntax for <Pattern> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the test pattern.
PRBS2E9, pattern type Pseudo Random Bit Sequence (PRBS) 2^9-1 is selected.
PRBS2E11, pattern type PRBS $2^{11}-1$ is selected.
PRBS2E15, pattern type PRBS $2^{15}-1$ is selected.
PRBS2E20, pattern type PRBS $2^{20}-1$ is selected.
PRBS2E23, pattern type PRBS $2^{23}-1$ is selected.
PRBS2E31, pattern type PRBS $2^{31}-1$ is selected.
P1100, pattern type 1100 is selected.
P1010, pattern type 1010 is selected.
P1111, pattern type 1111 is selected.
P0000, pattern type 0000 is selected.

:SENSe[1..n]:DATA:TELeom:PATTErn:TYPE?

P1IN8, 1 in 8 as pattern type is selected.
P1IN16, 1 in 16 as pattern type is selected.
P3IN24, 3 in 24 as pattern type is selected.
QRSS, Quasi-Random Signal Source (QRSS) pattern type is selected.
DALY, DALY pattern type is selected.
P55OCTET, 55OCTET pattern type is selected.
NCLIENT, Null Client pattern type is selected.
UPATTERN, User Pattern is selected.

Example(s)

* SENS:DATA:TEL:PATT:TYPE PRBS2E9
* SENS:DATA:TEL:PATT:TYPE? Returns PRBS2E9

See Also

* SENSe[1..n]:DATA:TELeom:PATTErn:TYPE

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

Description This command enables or disables transmitter coupling.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:PATtern:
COUPled <wsp> <Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element.
Enables or disables the transmitter coupling.

Example(s) * SOUR:DATA:TEL:PATT:COUP ON
* SOUR:DATA:TEL:PATT:COUP? Returns:1

See Also * SOURce[1..n]:DATA:TELEcom:PATTERN:
COUPled?

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

Description	<p>This query returns the status of the transmitter coupling.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:COUPled?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the coupled status of the transmitter.</p>
Example(s)	<p>* SOUR:DATA:TEL:PATT:COUP ON</p> <p>* SOUR:DATA:TEL:PATT:COUP? Returns:1</p>
See Also	<p>* SOUR:DATA:TEL:PATT:COUP ON</p> <p>* SOUR:DATA:TEL:PATT:COUP? Returns:1</p>

SONET/SDH SCPI Command Reference

SONET/SDH Analyzer

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

Description	<p>This command enables or disables the receiver coupling.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:PATtern:COUPled <wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Enables or disables the receiver coupling.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:PATT:COUP ON* SENS:DATA:TEL:PATT:COUP? Returns: 1
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:PATTERN:COUPled?

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

Description	This query returns the status of the receiver coupling. At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:PATtern:COUPled?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The program data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the coupled status of the receiver.
Example(s)	* SENS:DATA:TEL:PATT:COUP ON * SENS:DATA:TEL:PATT:COUP? Returns: 1
See Also	* SENSe[1..n]:DATA:TELEcom:PATtern: COUPled?

:SOURCE[1..n]:DATA:TELEcom:POLarity

Description	<p>This command sets the polarity of payload pattern for the transmitter.</p> <p>At *RST, this value is set to NINVerted.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:POLarity<wsp> NINVerted INVerted</pre>
Parameter(s)	<p>Polarity:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NINVerted INVerted.</p> <p>Selects the polarity pattern for the transmitter. NINVerted, selects the polarity as Non Inverted. INVerted, selects the polarity as Inverted.</p>
Example(s)	<pre>* SOUR:DATA:TEL:PATT:POL INV * SOUR:DATA:TEL:PATT:POL? Returns INVERTED</pre>
See Also	<pre>* SOURCE[1..n]:DATA:TELEcom:POLarity?</pre>

:SOURCE[1..n]:DATA:TELEcom:POLarity?

Description	<p>This query returns the polarity of payload pattern for the transmitter.</p> <p>At *RST, this value is set to NINVerted.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:POLarity?
Parameter(s)	None
Response Syntax	<Polarity>
Response(s)	<p>Polarity: The program data syntax for <Polarity> is defined as a <CHARACTER RESPONSE DATA> element. Returns the polarity pattern for the transmitter. NINVERTED, Non Inverted is selected as polarity pattern.</p>

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

INVERTED, Inverted is selected as polarity pattern.

Example(s)

* SOUR:DATA:TEL:POL INV

* SOUR:DATA:TEL:POL?

Returns INVERTED

See Also

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

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

Description	<p>This command sets the polarity of payload pattern for the receiver.</p> <p>At *RST, this value is set to NINVerted.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:POLarity<wsp> NINVerted INVerted</p>
Parameter(s)	<p>Polarity: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NINVerted INVerted. Selects the polarity pattern for the receiver. NINVerted, selects the polarity as Non Inverted. INVerted, selects the polarity as Inverted.</p>
Example(s)	<p>* SENS:DATA:TEL:POL INV * SENS:DATA:TEL:POL? Returns INVERTED</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:POLarity?</p>

:SENSe[1..n]:DATA:TELeom:POLarity?

Description	This query returns the polarity of payload pattern for the receiver. At *RST, this value is set to NINVerted.
Syntax	:SENSe[1..n]:DATA:TELeom:POLarity?
Parameter(s)	None
Response Syntax	<Polarity>
Response(s)	Polarity: The response data syntax for <Polarity> is defined as a <CHARACTER RESPONSE DATA> element. Returns the polarity pattern for the receiver. NINVERTED, Non Inverted is selected as polarity pattern. INVERTED, Inverted is selected as polarity pattern.
Example(s)	* SENS:DATA:TEL:POL INV * SENS:DATA:TEL:POL? Returns INVERTED
See Also	* SENSe[1..n]:DATA:TELeom:POLarity

:SOURce[1..n]:DATA:TELEcom:PATtern: OVERwrite:ENABled

Description	<p>This command enables or disables the Overwrite feature for pattern.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:PATtern: OVERwrite:ENABled <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the first parameter is defined as a <Boolean Program Data> element. Enables or disables the Overwrite feature for pattern.</p>
Example(s)	<pre>* SENS:DATA:TEL:PATT:OVER:ENAB ON * SENS:DATA:TEL:PATT:OVER:ENAB? Returns 1</pre>
See Also	<pre>* SENSE[1..n]:DATA:TELEcom:PATtern: OVERwrite:ENABled?</pre>

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

Description	<p>This query returns the status of Overwrite feature for pattern.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:PATtern: OVERwrite:ENABled?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.</p> <p>Returns the Overwrite feature for pattern.</p>
Example(s)	<p>* SENS:DATA:TEL:PATT:OVER:ENAB ON</p> <p>* SENS:DATA:TEL:PATT:OVER:ENAB? Returns 1</p>
See Also	* SENSE[1..n]:DATA:TELEcom:PATtern: OVERwrite:ENABled

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

Description	<p>This command sets the user pattern value of the specified index for transmitter.</p> <p>At *RST, this value is #H00000000.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE: USER:VALue <wsp> <Pattern> , <Value></p>
Parameter(s)	<p>Pattern: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the user pattern type for the transmitter. Choices are 1 through 10.</p>

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

Value:

The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the user pattern value of the specified index for transmitter.

Choices are #H00000000 through #HFFFFFFF.

Example(s)

```
* SOUR:DATA:TEL:PATT:TYPE UPAT
* SOUR:DATA:TEL:PATT:TYPE:USER:VAL
2,#HFFFF0000
* SOUR:DATA:TEL:PATT:TYPE:USER:VAL? 2
Returns #HFFFF0000
```

See Also

```
* 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?**

Description	<p>This query returns the user pattern value of the specified index for transmitter.</p> <p>At *RST, this value is #H00000000.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE: USER:VALue?<wsp><Pattern></p>
Parameter(s)	<p>Pattern: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the user pattern type for the transmitter. Choices are 1 through 10.</p>
Response Syntax	<p><Value></p>

**:SOURce[1..n]:DATA:TELEcom:PATtern:TYPE:
USER:VALue?****Response(s)**

Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the user pattern value of the specified index.

Example(s)

```
* SOUR:DATA:TEL:PATT:TYPE UPAT
```

```
* SOUR:DATA:TEL:PATT:TYPE:USER:VAL  
2,#HFFFF0000
```

```
* SOUR:DATA:TEL:PATT:TYPE:USER:VAL? 2
```

Returns #HFFFF0000

See Also

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

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

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

Description	<p>This command sets the user pattern value of the specified index for receiver.</p> <p>At *RST, this value is #H00000000.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:PATtern:TYPE: USER:VALue <wsp> <Pattern> , <Value></p>
Parameter(s)	<p>Pattern: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the user pattern type for receiver. Choices are 1 through 10.</p>

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

The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the user pattern value of the specified index for receiver.

Choices are #H00000000 through #HFFFFFFF.

Example(s)

```
* SENS:DATA:TEL:PATT:TYPE UPAT
* SENS:DATA:TEL:PATT:TYPE:USER:VAL
2,#HFFFF0000
* SENS:DATA:TEL:PATT:TYPE:USER:VAL? 2
Returns #HFFFF0000
```

See Also

```
* 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?**

Description This query returns the user pattern value of the specified index for receiver.

At *RST, this value is #H00000000.

Syntax :SENSe[1..n]:DATA:TELEcom:PATtern:TYPE:
USER:VALue?<wsp><Pattern>

Parameter(s) Pattern:
The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.
Sets the user pattern type for receiver.
Choices are 1 through 10.

Response Syntax <Pattern>

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

Response(s)	<p>Pattern: The response data syntax for <Pattern> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the user pattern value of the specified index.</p>
Example(s)	<pre>* SENS:DATA:TEL:PATT:TYPE UPAT * SENS:DATA:TEL:PATT:TYPE:USER:VAL 2,#HFFFF0000 * SENS:DATA:TEL:PATT:TYPE:USER:VAL? 2 Returns #HFFFF0000</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELeom:PATtern:TYPE * SENSe[1..n]:DATA:TELeom:PATtern:TYPE: USER:VALue</pre>

:SENSe[1..n]:DATA:TELEcom:PATtern:TRAFfic

Description This command sets the live traffic. When enabled, live traffic analyzes the line traffic without test pattern, thus squelching the pattern loss and bit error indication.

At *RST, this value is set to OFF.

Syntax :SENSe[1..n]:DATA:TELEcom:PATtern:TRAFfic
<wsp> <Set>

Parameter(s) Set:
The program data syntax for <Set> is defined as a <Boolean Program Data> element.
The <Set> special forms ON and OFF are accepted on input for increased readability.
ON corresponds to 1 and OFF corresponds to 0.
Enables or disables the live traffic.

Example(s) * SENS:DATA:TEL:PATT:TRAF ON
* SENS:DATA:TEL:PATT:TRAF? Returns 1

See Also * SENSe[1..n]:DATA:TELEcom:PATtern:TRAFfic?

:SENSe[1..n]:DATA:TELEcom:PATtern:TRAFfic?

Description	This query returns the status of live traffic. At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:PATtern:TRAFfic?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of live traffic.
Example(s)	* SENS:DATA:TEL:PATT:TRAF ON * SENS:DATA:TEL:PATT:TRAF? Returns 1
See Also	* SENSe[1..n]:DATA:TELEcom:PATtern:TRAFfic

**:SOURce[1..n]:DATA:TELEcom:PATTErn:ALARm:
PATTErn:TYPE**

Description	<p>This command sets the type of pattern alarm.</p> <p>At *RST, this value is set to PLOSSs.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PATTErn:ALARm: PATTErn:TYPE<wsp>PLOSSs</p>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> element for this parameter is: PLOSSs. Selects the type of pattern alarm. PLOSSs, selects the type of pattern alarm as Pattern Loss.</p>
Example(s)	<p>* SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE PLOSS * SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE? Returns PLOSS</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:PATTErn: ALARm:PATTErn:TYPE?</p>

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

Description	This query returns the type of pattern alarm. At *RST, this value is set to PLOSs.
Syntax	:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	Alarm: The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of pattern alarm. PLOSs, Pattern Loss is selected as pattern alarm.
Example(s)	* SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE PLOS * SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE? Returns PLOSS
See Also	* SOURce[1..n]:DATA:TELEcom:PATtern: ALARm:PATtern:TYPE

:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern

Description	<p>This command enables or disables the status of pattern alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern<wsp><Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE PLOS* SOUR:DATA:TEL:PATT:ALAR:PATT ON* SOUR:DATA:TEL:PATT:ALAR:PATT? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:TYPE* SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern?

:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern?

Description	<p>This query returns the status of pattern alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of pattern alarm generation.</p>
Example(s)	<p>* SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE PLOS</p> <p>* SOUR:DATA:TEL:PATT:ALAR:PATT ON</p> <p>* SOUR:DATA:TEL:PATT:ALAR:PATT? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:PATtern: ALARm:PATtern:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:PATtern: ALARm:PATtern</p>

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

Description	<p>This command selects the manual type of pattern error.</p> <p>At *RST, this value is set to BIT.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PATtern:ERror: PATtern:MANual:TYPE<wsp>BIT</p>
Parameter(s)	<p>Error: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. Selects the type of pattern error. BIT, selects the type of pattern error as Bit.</p>
Example(s)	<p>* SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT * SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE? Returns BIT</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:PATtern:ERror: PATtern:MANual:TYPE?</p>

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

Description	<p>This query returns the manual type of pattern error.</p> <p>At *RST, this value is set to BIT.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of pattern error.</p> <p>BIT, Bit is selected as pattern error.</p>
Example(s)	<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>
See Also	* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AMOut**

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

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AMOut <wsp> <Amount>
|MAXimum|MINimum

Parameter(s) Amount:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum|MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AMOut**

Sets the amount of pattern error to be injected.
Choices are 1 through 50.

Example(s)

- * SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT
- * SOUR:DATA:TEL:PATT:ERR:PATT:AMO 15
- * SOUR:DATA:TEL:PATT:ERR:PATT:AMO? Returns 15

See Also

- * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:MANual:TYPE
- * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AMOut?

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

Description	<p>This query returns the amount of pattern error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AMOUnt? [<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimumMAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<p><Amount></p>

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

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of pattern error injected into the instrument.
Example(s)	* SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT * SOUR:DATA:TEL:PATT:ERR:PATT:AMO 15 * SOUR:DATA:TEL:PATT:ERR:PATT:AMO? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AMOut

:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:INJect

Description	<p>This command injects the pattern error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:INJect</pre>
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT* SOUR:DATA:TEL:PATT:ERR:PATT:AMO 15* SOUR:DATA:TEL:PATT:ERR:PATT:INJ
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AMOUNT

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERROr:
PATtern:HISTory?**

Description	<p>This query returns the history status of pattern error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PATtern:ERROr: PATtern:HISTory?<wsp>BIT</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BIT.</p> <p>Selects the type of pattern error.</p> <p>BIT, selects the type of pattern error as Bit.</p>
Response Syntax	<pre><History></pre>

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:HISTory?**

Response(s)

History:
The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the history status of pattern 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.

Example(s)

* SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT
* SOUR:DATA:TEL:PATT:ERR:PATT:AMO 15
* SOUR:DATA:TEL:PATT:ERR:PATT:INJ
* FETC:DATA:TEL:PATT:ERR:PATT:HIST? BIT
Returns the error history.

See Also

* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AMOUNT
* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:INJect

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:SEConds?**

Description	<p>This query returns the number of seconds within which pattern error occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:SEConds?<wsp>BIT</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BIT.</p> <p>Selects the type of pattern error.</p> <p>BIT, selects the type of pattern error as Bit.</p>
Response Syntax	<pre><Seconds></pre>

:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:SEConds?

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of pattern error.
Example(s)	* SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT * SOUR:DATA:TEL:PATT:ERR:PATT:AMO 15 * SOUR:DATA:TEL:PATT:ERR:PATT:INJ * FETC:DATA:TEL:PATT:ERR:PATT:SEC? BIT Returns the number of errored seconds.
See Also	* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AMOUNT * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:INJect

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:CURRent?**

Description	<p>This query returns the current status of pattern error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:CURRent?<wsp>BIT</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BIT.</p> <p>Selects the type of pattern error.</p> <p>BIT, selects the type of pattern error as Bit.</p>
Response Syntax	<pre><Current></pre>

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:CURRent?**

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current status of pattern 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:PATT:ERR:PATT:MAN:TYPE BIT

* SOUR:DATA:TEL:PATT:ERR:PATT:AMO 15

* SOUR:DATA:TEL:PATT:ERR:PATT:INJ

* FETC:DATA:TEL:PATT:ERR:PATT:CURR? BIT

Returns the current error status.

See Also

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

* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AMOUNT

* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:INJect

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:COUNT?**

Description	<p>This query returns the count of pattern error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:COUNT? <wsp>BIT</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BIT.</p> <p>Selects the type of pattern error.</p> <p>BIT, selects the type of pattern error as Bit.</p>
Response Syntax	<pre><Count></pre>

:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:COUNT?

Response(s)	<p>Count:</p> <p>The program data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the count of pattern error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT* SOUR:DATA:TEL:PATT:ERR:PATT:AMO 15* SOUR:DATA:TEL:PATT:ERR:PATT:INJ* FETC:DATA:TEL:PATT:ERR:PATT:COUN? BIT <p>Returns the error count.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AMOut* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:INJect

**:FETCh[1..n]:DATA:TELEcom:PATtern:ERROr:
PATtern:RATE?**

Description	<p>This query returns the current error rate of pattern error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PATtern:ERROr: PATtern:RATE? <wsp>BIT</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BIT.</p> <p>Selects the type of pattern error.</p> <p>BIT, selects the type of pattern error as Bit.</p>
Response Syntax	<pre><Rate></pre>

:FETCh[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of pattern error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PATT:ERR:PATT:MAN:TYPE BIT* SOUR:DATA:TEL:PATT:ERR:PATT:AMO 15* SOUR:DATA:TEL:PATT:ERR:PATT:INJ* FETC:DATA:TEL:PATT:ERR:PATT:RATE? BIT <p>Returns the error rate.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AMOUnt* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:INJect

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

Description	<p>This query returns the history status of pattern alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern:HISTory?<wsp>PLOSs</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: PLOSs.</p> <p>Selects the type of pattern alarm.</p> <p>PLOSs, selects the type of pattern alarm as Pattern Loss.</p>
Response Syntax	<pre><History></pre>

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

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE PLOS* SOUR:DATA:TEL:PATT:ALAR:PATT ON* FETC:DATA:TEL:PATT:ALAR:PATT:HIST? PLOS <p>Returns the alarm history.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:TYPE* SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern

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

Description	<p>This query returns the number of seconds within which pattern alarm occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern:SEConds?<wsp>PLOSs</p>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> element for this parameter is: PLOSs. Selects the type of pattern alarm. PLOSs, selects the type of pattern alarm as Pattern Loss.</p>
Response Syntax	<p><Seconds></p>

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

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of pattern alarm.
Example(s)	* SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE PLOS * SOUR:DATA:TEL:PATT:ALAR:PATT ON * FETC:DATA:TEL:PATT:ALAR:PATT:SEC? PLOS Returns the number of seconds of pattern alarm.
See Also	* SOURce[1..n]:DATA:TELEcom:PATtern: ALARm:PATtern:TYPE * SOURce[1..n]:DATA:TELEcom:PATtern: ALARm:PATtern

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

Description	<p>This query returns the current status of pattern alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:PATtern:ALARm: PATtern:CURRent?<wsp>PLOSs</p>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> element for this parameter is: PLOSs. Selects the type of pattern alarm. PLOSs, selects the type of pattern alarm as Pattern Loss.</p>
Response Syntax	<p><Current></p>

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

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns 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 is not running.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PATT:ALAR:PATT:TYPE PLOS* SOUR:DATA:TEL:PATT:ALAR:PATT ON* FETC:DATA:TEL:PATT:ALAR:PATT:CURR? PLOS <p>Returns the current alarm status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern:TYPE* SOURce[1..n]:DATA:TELEcom:PATtern:ALARm:PATtern

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

Description	<p>This command sets the type of Pattern error for automated injection.</p> <p>At *RST, this value is set to BIT.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:TYPE<wsp>BIT</pre>

:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:TYPE

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter is: BIT.</p> <p>Selects the type of pattern error for automated injection.</p> <p>BIT, selects the type of pattern error as Bit.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE BIT* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE? Returns BIT
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE?* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated

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

Description	This query returns the type of pattern error for automated injection. At *RST, this value is set to BIT.
Syntax	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

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

Response(s)	Error: The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Pattern error for the automated injection. BIT, Bit is selected as pattern error.
Example(s)	* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE BIT * SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE? Returns BIT
See Also	* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated:RATE**

Description	<p>This command sets the injection rate for the selected Pattern error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:RATE<wsp><Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated:RATE**

Sets the injection rate for the selected pattern error.

Example(s)

```
* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE BIT
* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:RATE
1.0E-05
* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:RATE?
Returns 1.0E-05
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated:RATE?
* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated
```


**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected pattern error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:RATE? [<wsp> MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injected rate will be returned.</p>
Response Syntax	<pre><Rate></pre>

:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected pattern error.</p>
Example(s)	<p>* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE BIT</p> <p>* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:RATE 1.0E-05</p> <p>* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:RATE? Returns 1.0E-05</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated****Description**

This command enables or disables the selected automated Pattern error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated**

Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated Pattern error injection.
Example(s)	* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE BIT * SOUR:DATA:TEL:PATT:ERR:PATT:AUT:RATE 1.0E-05 * SOUR:DATA:TEL:PATT:ERR:PATT:AUT ON * SOUR:DATA:TEL:PATT:ERR:PATT:AUT? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated?

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

Description	This query returns the status of automated pattern error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

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

Response(s)	Set The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated Pattern error injection.
Example(s)	* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE BIT * SOUR:DATA:TEL:PATT:ERR:PATT:AUT:RATE 1.0E-05 * SOUR:DATA:TEL:PATT:ERR:PATT:AUT ON * SOUR:DATA:TEL:PATT:ERR:PATT:AUT? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:PATtern:AUTomated

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated:CONTInuous****Description**

This command enables or disables the continuous rate of automated pattern error injection.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated:CONTInuous <wsp>
<Set>

:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:CONTInuous

Parameter(s)

Set:

The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated Pattern error injection continuously.

Example(s)

* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE BIT
* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:CONT ON
* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:CONT?
Returns 1
* SOUR:DATA:TEL:PATT:ERR:PATT:AUT ON

See Also

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

**:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated:CONTInuous?**

Description	This query returns the status of continuous rate of automated pattern error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:PATtern:ERRor: PATtern:AUTomated:CONTInuous?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of continuous rate of automated Pattern error injection.

Example(s)

* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:TYPE BIT

* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:CONT ON

* SOUR:DATA:TEL:PATT:ERR:PATT:AUT:CONT?

Returns 1

* SOUR:DATA:TEL:PATT:ERR:PATT:AUT ON

See Also

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

* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated

* SOURce[1..n]:DATA:TELEcom:PATtern:ERRor:
PATtern:AUTomated:CONTInuous

**:FETCh[1..n]:DATA:TELEcom:PATtern:PM:
STATistics?****Description**

This query returns the performance monitoring statistics of Pattern.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:PATtern:PM:
STATistics?<wsp>G821|M2100OOSM, EFS|EC|
ES|SES|UAS|ESR|SESR|DM

Parameter(s)

Standard:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
G821|M2100OOSM.

Selects the performance monitoring standard number.

G821, selects G.821 as a standard number.

M2100OOSM, selects M.2100 OOSM as a standard number.

:FETCh[1..n]:DATA:TELEcom:PATtern:PM: STATistics?

Type:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

EFS|EC|ES|SES|UAS|ESR|SESR|DM.

Selects the performance monitoring statistics.

EFS, selects Error Free Seconds (EFS).

EC, selects Error Count (EC).

ES, selects Errored Seconds (ES).

SES, selects Severely Errored Seconds (SES).

UAS, selects Unavailable Second (UAS).

ESR, selects Errored Second Ratio (ESR).

SESR, selects Severely Errored Second Ratio (SESR).

DM, selects Degraded Minutes (DM).

**:FETCh[1..n]:DATA:TELEcom:PATtern:PM:
STATistics?**

Response Syntax <Statistics>

Response(s) Statistics:
The response data syntax for <Statistics> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns performance monitoring statistics of Pattern.

Example(s) * FETC:DATA:TEL:PATT:PM:STAT? G821,EFS
Returns performance monitoring statistics of Pattern.

SONET Command Reference

:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:ENABLEd

Description This command enables or disables the J0 Trace. When enabled, generates the J0 Trace message defined to give access to the trace format and message. When the J0 Trace is disabled, the J0 1-byte format is used.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:ENABLEd<wsp><Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element.
The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the J0 trace.

Example(s) * SOUR:DATA:TEL:SON:OVER:J:ENAB ON
* SOUR:DATA:TEL:SON:OVER:J:ENAB? Returns 1

See Also * SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:ENABLEd?

**:SOURCE[1..n]:DATA:TELEcom:SONet:
OVERhead:J:ENABLEd?**

Description	This query returns the status of the J0 trace. At *RST, this value is set to OFF.
Syntax	:SOURCE[1..n]:DATA:TELEcom:SONet: OVERhead:J:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns 0 if J0 trace is disabled, or returns 1.
Example(s)	* SOUR:DATA:TEL:SON:OVER:J:ENAB ON * SOUR:DATA:TEL:SON:OVER:J:ENAB? Returns 1
See Also	* SOURCE[1..n]:DATA:TELEcom:SONet: OVERhead:J:ENABLEd

:SOURCE[1..n]:DATA:TELEcom:SONet: OVERhead:J:OVERwrite:ENABLEd

Description This command enables or disables the Overwrite feature.

At *RST, this value is set to OFF.

Syntax :SOURCE[1..n]:DATA:TELEcom:SONet:
OVERhead:J:OVERwrite:ENABLEd <wsp> <Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Overwrite feature.

**:SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:OVERwrite:ENABLEd**

Example(s)	* SOUR:DATA:TEL:SON:OVER:J:ENAB ON * SOUR:DATA:TEL:SON:OVER:J:OVER:ENAB ON * SOUR:DATA:TEL:SON:OVER:J:OVER:ENAB? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:ENABLEd * SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:OVERwrite:ENABLEd?

:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:OVERwrite:ENABled?

Description	<p>This query returns the status of the Overwrite feature.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:OVERwrite:ENABled?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Overwrite feature.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:OVERwrite:ENABLEd?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:OVER:J:ENAB ON* SOUR:DATA:TEL:SON:OVER:J:OVER:ENAB ON* SOUR:DATA:TEL:SON:OVER:J:OVER:ENAB? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:ENABLEd* SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:OVERwrite:ENABLEd

:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern

Description	<p>This command sets the J0 value in 16 or 64 bytes format.</p> <p>At *RST, this value is set to B16.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern<wsp>B16 B64</p>
Parameter(s)	<p>Pattern: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: B16 B64. Sets the J0 value in 16 or 64 bytes format.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern**

B16, selects the 16 bytes format.

B64, selects the 64 bytes format.

Example(s)

* SOUR:DATA:TEL:SON:OVER:J:ENAB ON

* SOUR:DATA:TEL:SON:OVER:J:PATT B16

* SOUR:DATA:TEL:SON:OVER:J:PATT?

Returns B16

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:ENABLEd

* SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern?

**:SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern?**

Description	<p>This query returns the J0 value in 16 or 64 bytes format.</p> <p>At *RST, this value is set to B16.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern?</code>
Parameter(s)	None
Response Syntax	<Pattern>
Response(s)	<p>Pattern:</p> <p>The response data syntax for <Pattern> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the J0 value in 16 or 64 bytes format.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern?**

B16, 16 bytes format is selected.

B64, 64 bytes format is selected.

Example(s)

* SOUR:DATA:TEL:SON:OVER:J:ENAB ON

* SOUR:DATA:TEL:SON:OVER:J:PATT B16

* SOUR:DATA:TEL:SON:OVER:J:PATT?

Returns B16

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:ENABLEd

* SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern

**:SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern:B16**

Description	<p>This command sets the 16 bytes format of the J0 trace.</p> <p>At *RST, this value is set to "EXFO SONET/SDH".</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern:B16<wsp><Message></p>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the 16 bytes format for J0 trace.</p>

**:SOURCE[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern:B16****Example(s)**

- * SOUR:DATA:TEL:SON:OVER:J:ENAB ON
- * SOUR:DATA:TEL:SON:OVER:J:PATT B16
- * SOUR:DATA:TEL:SON:OVER:J:PATT:B16 "EXFO SONET/SDH"
- * SOUR:DATA:TEL:SON:OVER:J:PATT:B16?

Returns "EXFO SONET/SDH"**Note**

16 bytes allows typing up to 15 bytes.

See Also

- * SOURCE[1..n]:DATA:TELEcom:SONet:OVERhead:J:ENABLEd
- * SOURCE[1..n]:DATA:TELEcom:SONet:OVERhead:J:PATtern
- * SOURCE[1..n]:DATA:TELEcom:SONet:OVERhead:J:PATtern:B16?

**:SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern:B16?**

Description	This query returns the 16 bytes format of the J0 trace. At *RST, this value is set to "EXFO SONET/SDH".
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern:B16?
Parameter(s)	None
Response Syntax	<Message>

**:SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern:B16?****Response(s)**

Message:

The program data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the 16 bytes format for J0 trace.

Example(s)

```
* SOUR:DATA:TEL:SON:OVER:J:ENAB ON
* SOUR:DATA:TEL:SON:OVER:J:PATT B16
* SOUR:DATA:TEL:SON:OVER:J:PATT:B16 "EXFO
SONET/SDH"
* SOUR:DATA:TEL:SON:OVER:J:PATT:B16?
Returns "EXFO SONET/SDH"
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:ENABled
* SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern
* SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern:B16
```

:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern:B64

Description	<p>This command sets the 64 bytes format of the J0 trace.</p> <p>At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern:B64<wsp><Message></p>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the 64 bytes format of J0 trace.</p>

**:SOURCE[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern:B64**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:OVER:J:ENAB ON* SOUR:DATA:TEL:SON:OVER:J:PATT B64* SOUR:DATA:TEL:SON:OVER:J:PATT:B64 "EXFO SONET/SDH Analyzer Section/RS trace test message"* SOUR:DATA:TEL:SON:OVER:J:PATT:B64? Returns "EXFO SONET/SDH Analyzer Section/RS trace test message"
Note	64 bytes allows typing up to 62 bytes.
See Also	<ul style="list-style-type: none">* SOURCE[1..n]:DATA:TELEcom:SONet:OVERhead:J:ENABled* SOURCE[1..n]:DATA:TELEcom:SONet:OVERhead:J:PATtern* SOURCE[1..n]:DATA:TELEcom:SONet:OVERhead:J:PATtern:B64?

:SOURCE[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern:B64?

Description	This query returns the 64 bytes format of the J0 trace. At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".
Syntax	:SOURCE[1..n]:DATA:TELEcom:SONet: OVERhead:J:PATtern:B64?
Parameter(s)	None
Response Syntax	<Message>

**:SOURCE[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern:B64?****Response(s)**

Message:

The program data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the 64 bytes format of J0 trace.

Example(s)

* SOUR:DATA:TEL:SON:OVER:J:ENAB ON
* SOUR:DATA:TEL:SON:OVER:J:PATT B64
* SOUR:DATA:TEL:SON:OVER:J:PATT:B64 "EXFO
SONET/SDH Analyzer Section/RS trace test
message"
* SOUR:DATA:TEL:SON:OVER:J:PATT:B64?
Returns "EXFO SONET/SDH Analyzer Section/RS
trace test message"

See Also

* SOURCE[1..n]:DATA:TELEcom:SONet:
OVERhead:J:ENABLEd
* SOURCE[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern
* SOURCE[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern:B64

:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM

Description	<p>This command enables or disables the Trace Identifier Mismatch (TIM) of J0 trace for the expected message defined.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the TIM (Trace Identifier Mismatch).</p>
Example(s)	<pre>* SENS:DATA:TEL:SON:OVER:J:TIM ON * SENS:DATA:TEL:SON:OVER:J:TIM? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:SONet:OVERhead :J:TIM?</pre>

**:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:
J:TIM?**

Description	This query returns the status of Trace Identifier Mismatch (TIM) for the expected message defined. At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the TIM (Trace Identifier Mismatch).
Example(s)	* SENS:DATA:TEL:SON:OVER:J:TIM ON * SENS:DATA:TEL:SON:OVER:J:TIM? Returns 1
See Also	* SENSe[1..n]:DATA:TELEcom:SONet:OVERhead :J:TIM

:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM:PATtern

Description	<p>This command sets the Trace Identifier Mismatch (TIM) value of J0 trace in 16 or 64 bytes format.</p> <p>At *RST, this value is set to B16.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM:PATtern<wsp>B16 B64</p>
Parameter(s)	<p>Pattern: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: B16 B64. Sets the format for J0 trace.</p>

**:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:
J:TIM:PATtern**

B16, selects the 16 Byte format.

B64, selects the 64 Byte format.

Example(s)

* SENS:DATA:TEL:SON:OVER:J:TIM ON

* SENS:DATA:TEL:SON:OVER:J:TIM:PATT B16

* SENS:DATA:TEL:SON:OVER:J:TIM:PATT?

Returns B16

See Also

* SENSe[1..n]:DATA:TELEcom:SONet:OVERhead
:J:TIM

* SENSe[1..n]:DATA:TELEcom:SONet:OVERhead
:J:TIM:PATtern?

:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM:PATtern?

Description	<p>This query returns the Trace Identifier Mismatch (TIM) value of J0 trace in 16 or 64 bytes format.</p> <p>At *RST, this value is set to B16.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM:PATtern?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Pattern></p>
Response(s)	<p>Pattern: The response data syntax for <Pattern> is defined as a <CHARACTER RESPONSE DATA> element. Returns the format for J0 trace.</p>

**:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:
J:TIM:PATtern?**

B16, 16 Byte format is selected.

B64, 64 Byte format is selected.

Example(s)

* SENS:DATA:TEL:SON:OVER:J:TIM ON

* SENS:DATA:TEL:SON:OVER:J:TIM:PATT B16

* SENS:DATA:TEL:SON:OVER:J:TIM:PATT?

Returns B16

See Also

* SENSe[1..n]:DATA:TELEcom:SONet:OVERhead
:J:TIM

* SENSe[1..n]:DATA:TELEcom:SONet:OVERhead
:J:TIM:PATtern

:FETCh[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM:PATtern:RECeived?

Description	This query returns the received J0 value in 16 or 64 bytes format. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM:PATtern:RECeived?
Parameter(s)	None
Response Syntax	<Message>

**:FETCh[1..n]:DATA:TELEcom:SONet:OVERhead:
J:TIM:PATtern:RECeived?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the received J0 value.

Example(s)

* SOUR:DATA:TEL:SON:OVER:J:ENAB ON

* SOUR:DATA:TEL:SON:OVER:J:PATT B16

* FETC:DATA:TEL:SON:OVER:J:TIM:PAT:REC?

Returns the J0 value in 16 or 64 bytes format.

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:ENABLEd

* SOURce[1..n]:DATA:TELEcom:SONet:
OVERhead:J:PATtern

:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM:PATtern:B16

Description	<p>This command sets the expected message for 16 bytes format of J0 trace.</p> <p>At *RST, this value is set to "EXFO SONET/SDH".</p>
Syntax	<code>:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM:PATtern:B16<wsp> <Message></code>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the expected message for 16 bytes format.</p>
Example(s)	<pre>* SENS:DATA:TEL:SON:OVER:J:TIM ON * SENS:DATA:TEL:SON:OVER:J:TIM:PATT B16 * SENS:DATA:TEL:SON:OVER:J:TIM:PATT:B16 "EXFO SONET/SDH" * SENS:DATA:TEL:SON:OVER:J:TIM:PATT:B16? Returns "EXFO SONET/SDH"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:SONet:OVERhead :J:TIM * SENSe[1..n]:DATA:TELEcom:SONet:OVERhead :J:TIM:PATtern * SENSe[1..n]:DATA:TELEcom:SONet:OVERhead :J:TIM:PATtern:B16?</pre>

**:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:
J:TIM:PATtern:B16?**

Description	This query returns the expected message for 16 bytes format of J0 trace. At *RST, this value is set to "EXFO SONET/SDH".
Syntax	:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM:PATtern:B16?
Parameter(s)	None
Response Syntax	<Message>

:SENSe[1..n]:DATA:TELecom:SONet:OVERhead: J:TIM:PATtern:B16?

Response(s)

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the 16 bytes sequence of the J0 trace of the expected message.

Example(s)

```
* SENS:DATA:TEL:SON:OVER:J:TIM ON
* SENS:DATA:TEL:SON:OVER:J:TIM:PATT B16
* SENS:DATA:TEL:SON:OVER:J:TIM:PATT:B16
"EXFO SONET/SDH"
* SENS:DATA:TEL:SON:OVER:J:TIM:PATT:B16?
Returns "EXFO SONET/SDH"
```

See Also

```
* SENSe[1..n]:DATA:TELecom:SONet: OVERhead
:J:TIM
* SENSe[1..n]:DATA:TELecom:SONet: OVERhead
:J:TIM:PATtern
* SENSe[1..n]:DATA:TELecom:SONet: OVERhead
:J:TIM:PATtern:B16
```

**:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:
J:TIM:PATtern:B64**

Description This command sets the expected message for 64 bytes format of J0 trace.

At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".

Syntax :SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:
J:TIM:PATtern:B64<wsp> <Message>

:SENSe[1..n]:DATA:TELecom:SONet:OVERhead: J:TIM:PATTerN:B64

Parameter(s)	<p>Message:</p> <p>The program data syntax for <Message> is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the expected message for 64 bytes format.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SON:OVER:J:TIM ON* SENS:DATA:TEL:SON:OVER:J:TIM:PATT B64* SENS:DATA:TEL:SON:OVER:J:TIM:PATT:B64 <p>"EXFO SONET/SDH Analyzer Section/RS trace test message"</p> <ul style="list-style-type: none">* SENS:DATA:TEL:SON:OVER:J:TIM:PATT:B64? <p>Returns "EXFO SONET/SDH Analyzer Section/RS trace test message"</p>
See Also	<ul style="list-style-type: none">* SENSE[1..n]:DATA:TELecom:SONet:OVERhead:J:TIM* SENSE[1..n]:DATA:TELecom:SONet:OVERhead:J:TIM:PATTerN* SENSE[1..n]:DATA:TELecom:SONet:OVERhead:J:TIM:PATTerN:B64?

**:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead:
J:TIM:PATtern:B64?**

Description	This query returns the expected message for 64 bytes format. At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".
Syntax	:SENSe[1..n]:DATA:TELEcom:SONet:OVERhead: J:TIM:PATtern:B64?
Parameter(s)	None
Response Syntax	<Message>

:SENSe[1..n]:DATA:TELecom:SONet:OVERhead: J:TIM:PATtern:B64?

Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the expected message for 64 bytes format.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SON:OVER:J:TIM ON* SENS:DATA:TEL:SON:OVER:J:TIM:PATT B64* SENS:DATA:TEL:SON:OVER:J:TIM:PATT:B64 <p>"EXFO SONET/SDH Analyzer Section/RS trace test message"</p> <ul style="list-style-type: none">* SENS:DATA:TEL:SON:OVER:J:TIM:PATT:B64? <p>Returns "EXFO SONET/SDH Analyzer Section/RS trace test message"</p>
See Also	<ul style="list-style-type: none">* SENSE[1..n]:DATA:TELecom:SONet:OVERhead:J:TIM* SENSE[1..n]:DATA:TELecom:SONet:OVERhead:J:TIM:PATtern* SENSE[1..n]:DATA:TELecom:SONet:OVERhead:J:TIM:PATtern:B64

:SOURce[1..n]:DATA:TELEcom:SONet:HOP: PATH:LABel

Description This command sets the path signal label (C2) of High Order Path (HOP) for the transmitter.

At *RST, the configuration is set to a device-dependent value.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:HOP:
PATH:LABel<wsp>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

Parameter(s) Label:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> 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

:SOURce[1..n]:DATA:TELEcom:SONet:HOP: PATH:LABel

|S11W|S12W|S13W|S14W|S15W|S16W|S17W
|S18W|S19W|S20W|S21W|S22W|S23W|S24W
|S25W|S26W|S27W|S28W|TSIGnal|AISTcm.

Selects the path signal label.

UNEQuipped, selects the unequipped path signal label.

EQUipped, selects the equipped non-specific path signal label.

FVTMode, selects the FVTMode VT (Virtual Tributary) Mode path signal label.

LOCKed, selects the locked VT (Virtual Tributary) Mode path signal label.

AMDS3, selects the Asynchronous Mapping for DS3 (Digital Signal-level 3) path signal label.

MDEvelopment, selects the mapping under development path signal label.

REServed, selects the reserved path signal label.

AM140, selects the Asynchronous Mapping for 140M (DS4NA) path signal label.

ATMM, selects the mapping for ATM path signal label.

MDQDb, selects the mapping for DQDB (Distributed Queue Dual Bus) path signal label.

FDDim, selects the Asynchronous Mapping for FDDI (Fiber Distributed Data Interface) path signal label.

MHDLC, selects the mapping of HDLC (High-Level Data Link Control) over SONET path signal label.

SSELf, selects the SSELf with self-synchronization scrambler path signal label.

**:SOURce[1..n]:DATA:TELEcom:SONet:HOP:
PATH:LABel**

MHLaps, selects the mapping of HDLC (High-Level Data Link Control)/LAPS (Link Access Procedure for SDH) path signal label.

SSET, selects the set SDL with use of a set-reset scrambler path signal label.

M10ETHERNET, selects the 10 Gbps ethernet (IEEE 802.3) path signal label.

GFP, selects the Generic Framing Procedure (GFP) path signal label.

RHPPp, selects the reserved [Obsolete High-Level Data Link Control (HDLC)/Point-to-Point Protocol (PPP) framed] path signal label.

S1W, selects the STS-1 (Synchronous Transport Signal-Level 1) w/1 VT (Virtual Tributary)x payload defect.

S2W, selects the STS-1 w/2 VT x payload defect.

S3W, selects the STS-1 w/3 VT x payload defect.

S4W, selects the STS-1 w/4 VT x payload defect.

S5W, selects the STS-1 w/5 VT x payload defect.

S6W, selects the STS-1 w/6 VT x payload defect.

S7W, selects the STS-1 w/7 VT x payload defect.

S8W, selects the STS-1 w/8 VT x payload defect.

S9W, selects the STS-1 w/9 VT x payload defect.

S10W, selects the STS-1 w/10 VT x payload defect.

S11W, selects the STS-1 w/11 VT x payload defect.

S12W, selects the STS-1 w/12 VT x payload defect.

S13W, selects the STS-1 w/13 VT x payload defect.

:SOURce[1..n]:DATA:TELEcom:SONet:HOP: PATH:LABel

S14W, selects the STS-1 w/14 VT x payload defect.

S15W, selects the STS-1 w/15 VT x payload defect.

S16W, selects the STS-1 w/16 VT x payload defect.

S17W, selects the STS-1 w/17 VT x payload defect.

S18W, selects the STS-1 w/18 VT x payload defect.

S19W, selects the STS-1 w/19 VT x payload defect.

S20W, selects the STS-1 w/20 VT x payload defect.

S21W, selects the STS-1 w/21 VT x payload defect.

S22W, selects the STS-1 w/22 VT x payload defect.

S23W, selects the STS-1 w/23 VT x payload defect.

S24W, selects the STS-1 w/24 VT x payload defect.

S25W, selects the STS-1 w/25 VT x payload defect.

S26W, selects the STS-1 w/26 VT x payload defect.

S27W, selects the STS-1 w/27 VT x payload defect.

S28W, selects the STS-1 w/28 VT x payload defect.

TSIGnal, selects the test signal, ITU-T 0.181 specific mapping path signal label.

**:SOURce[1..n]:DATA:TELEcom:SONet:HOP:
PATH:LABel**

AISTcm, selects the STS (Synchronous Transport Signal) SPE (Synchronous Payload Envelope) AIS (Alarm Indication Signal) (TCM) path signal label.

Example(s)

* SOUR:DATA:TEL:SON:HOP:PATH:LAB EQU
* SOUR:DATA:TEL:SON:HOP:PATH:LAB?
Returns EQUIPPED

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:HOP:
PATH:LABel?

:SOURce[1..n]:DATA:TELEcom:SONet:HOP: PATH:LABel?

Description	<p>This query returns the path signal label (C2) of High Order Path (HOP) for the transmitter.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:HOP: PATH:LABel?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Label></p>
Response(s)	<p>Label: The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element. Returns the path signal label. UNEQUIPPED, Unequipped path signal label is selected. EQUIPPED, Equipped non-specific path signal label is selected. FVTMODE, FVTMode Virtual Tributary (VT) Mode path signal label is selected. LOCKED, Locked Virtual Tributary (VT) Mode path signal label is selected.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:HOP:
PATH:LABel?**

AMDS3, Asynchronous Mapping for Digital Signal-level 3 (DS3) path signal label is selected.

MDEVELOPMENT, Mapping under development path signal label is selected.

RESERVED, Reserved path signal label is selected.

AM140, Asynchronous Mapping for 140M (DS4NA) path signal label.

ATMM, Mapping for ATM path signal label.

MDQDB, Mapping for DQDB path signal label.

FDDIM, Asynchronous Mapping for Fiber Distributed Data Interface (FDDI) path signal label.

MHDLc, Mapping of High-Level Data Link Control (HDLC) over SONET path signal label.

SSELF, SSELF with self-synchronization scrambler path signal label is selected.

MHLAPS, Mapping of High-Level Data Link Control (HDLC)/Link Access Procedure for SDH (LAPS) path signal label is selected.

SSET, Set SDL with use of a set-reset scrambler path signal label is selected.

M10ETHERNET, 10 Gbps ethernet (IEEE 802.3) path signal label is selected.

GFP, Generic Framing Procedure (GFP) path signal label is selected.

RHPPP, Reserved [Obsolete High-Level Data Link Control (HDLC)/Point-to-Point Protocol (PPP) framed] path signal label is selected.

S1W, Synchronous Transport Signal-Level 1 (STS-1) w/1 Virtual Tributary (VTx) payload defect is selected.

**:SOURce[1..n]:DATA:TELEcom:SONet:HOP:
PATH:LABel?**

S2W, STS-1 w/2 VTx payload defect is selected.
S3W, STS-1 w/3 VTx payload defect is selected.
S4W, STS-1 w/4 VTx payload defect is selected.
S5W, STS-1 w/5 VTx payload defect is selected.
S6W, STS-1 w/6 VTx payload defect is selected.
S7W, STS-1 w/7 VTx payload defect is selected.
S8W, STS-1 w/8 VTx payload defect is selected.
S9W, STS-1 w/9 VTx payload defect is selected.
S10W, STS-1 w/10 VTx payload defect is selected.
S11W, STS-1 w/11 VTx payload defect is selected.
S12W, STS-1 w/12 VTx payload defect is selected.
S13W, STS-1 w/13 VTx payload defect is selected.
S14W, STS-1 w/14 VTx payload defect is selected.
S15W, STS-1 w/15 VTx payload defect is selected.
S16W, STS-1 w/16 VTx payload defect is selected.
S17W, STS-1 w/17 VTx payload defect is selected.
S18W, STS-1 w/18 VTx payload defect is selected.
S19W, STS-1 w/19 VTx payload defect is selected.
S20W, STS-1 w/20 VTx payload defect is selected.
S21W, STS-1 w/21 VTx payload defect is selected.
S22W, STS-1 w/22 VTx payload defect is selected.
S23W, STS-1 w/23 VTx payload defect is selected.
S24W, STS-1) w/24 VTx payload defect is selected.
S25W, STS-1 w/25 VTx payload defect is selected.
S26W, STS-1 w/26 VTx payload defect is selected.
S27W, STS-1 w/27 VTx payload defect is selected.
S28W, STS-1 w/28 VTx payload defect is selected.

**:SOURce[1..n]:DATA:TELEcom:SONet:HOP:
PATH:LABel?**

TSIGNAL, Test signal, ITU-T 0.181 specific mapping path signal label is selected.

AISTCM, Synchronous Transport Signal (STS)

Synchronous Payload Envelope (SPE) Alarm

Indication Signal (AIS) (TCM) path signal label is selected.

Example(s)

* SOUR:DATA:TEL:SON:HOP:PATH:LAB EQU

* SOUR:DATA:TEL:SON:HOP:PATH:LAB?

Returns EQUIPPED

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:HOP:
PATH:LABel

:SENSe[1..n]:DATA:TELEcom:SONet:HOP:PATH: LABel:EXPeCted

Description	<p>This command sets the expected path signal label (C2) of High Order Path (HOP) for the receiver.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SONet:HOP:PATH: LABel:EXPeCted<wsp>EQUipped FVTMode LOCKed AMDS3 MDEvelopment AM140 ATMM MDQDb FDDim MHDLC SSElf MHLaps SSET M10ETHERNET GFP RHPPp TSIGNal</pre>
Parameter(s)	<p>Label:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>EQUipped FVTMode LOCKed AMDS3 MDEvelopment AM140 ATMM MDQDb FDDim MHDLC SSElf MHLaps SSET M10ETHERNET GFP RHPPp TSIGNal.</pre> <p>Selects the expected path signal label.</p> <p>EQUipped, selects the Equipped non-specific path signal label.</p>

**:SENSE[1..n]:DATA:TELEcom:SONet:HOP:PATH:
LABel:EXPEcted**

FVTMode, selects the FVTMode VT (Virtual Tributary) mode path signal label.

LOCKed, selects the Locked VT (Virtual Tributary) mode path signal label.

AMDS3, selects the Asynchronous Mapping for DS3 (Digital Signal-level 3) path signal label.

MDEvelopment, selects the Mapping under Development path signal label.

AM140, selects the Asynchronous Mapping for 140M (DS4NA) path signal label.

ATMM, selects the Mapping for ATM (Asynchronous Transfer Mode) path signal label.

MDQDb, selects the Mapping for DQDB (Distributed Queue Dual Bus) path signal label.

FDDim, selects the Asynchronous Mapping for Fiber Distributed Data Interface (FDDI) path signal label.

MHDLC, selects the Mapping of HDLC (High-Level Data Link Control) over SONET path signal label.

SSElf, selects the SSElf with self-synchronization scrambler path signal label.

MHLaps, selects the Mapping of HDLC (High-Level Data Link Control)/LAPS (Link Access Procedure for SDH) path signal label.

SSET, selects the Set SDL with use of a set-reset scrambler path signal label.

M10ETHERNET, selects the 10 Gbps ethernet (IEEE 802.3) path signal label.

GFP, selects the GFP (Generic Framing Procedure) path signal label.

**:SENSe[1..n]:DATA:TELecom:SONet:HOP:PATH:
LABel:EXPeCted**

RHPPp, selects the Reserved [obsolete HDLC (High-Level Data Link Control)/PPP (Point-to-Point Protocol) framed] path signal label.

TSIGnal, selects the Test signal, ITU-T 0.181 specific mapping path signal label.

Example(s)

* SENS:DATA:TEL:SON:HOP:PATH:LAB:EXP EQU
* SENS:DATA:TEL:SON:HOP:PATH:LAB:EXP?
Returns EQUIPPED

See Also

* SENSe[1..n]:DATA:TELecom:SONet:HOP:
PATH:LABel:EXPeCted?

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

Description	<p>This query returns the expected path signal label of High Order Path (HOP) for the receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:SONet:HOP:PATH: LABel:EXPEcted?
Parameter(s)	None
Response Syntax	<Label>
Response(s)	<p>Label:</p> <p>The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element. Returns the expected path signal label.</p> <p>EQUIPPED, Equipped non-specific path signal label is selected.</p> <p>FVTMODE, FVTMode Virtual Tributary (VT) mode path signal label is selected.</p> <p>LOCKED, Locked Virtual Tributary (VT) mode path signal label is selected.</p> <p>AMDS3, Asynchronous Mapping for Digital Signal-level 3 (DS3) path signal label is selected.</p> <p>MDEVELOPMENT, Mapping under development path signal label is selected.</p>

:SENSe[1..n]:DATA:TELEcom:SONet:HOP:PATH: LABel:EXPeCted?

AM140, Asynchronous Mapping for 140M (DS4NA) path signal label is selected.

ATMM, Mapping for ATM path signal label is selected.

MDQDB, Mapping for DQDB path signal label is selected.

FDDIM, Asynchronous Mapping for Fiber Distributed Data Interface (FDDI) path signal label is selected.

MHDLC, Mapping of High-Level Data Link Control (HDLC) over SONET path signal label is selected.

SSELF, SSELF with self-synchronization scrambler path signal label is selected.

MHLAPS, Mapping of High-Level Data Link Control (HDLC)/Link Access Procedure for SDH (LAPS) path signal label is selected.

SSET, Set SDL with use of a set-reset scrambler path signal label is selected.

M10ETHERNET, 10 Gbps ethernet (IEEE 802.3) path signal label is selected.

GFP, Generic Framing Procedure (GFP) path signal label is selected.

RHPPP, Reserved [obsolete High-Level Data Link Control (HDLC)/Point-to-Point Protocol (PPP) framed] path signal label is selected.

TSIGNAL, Test signal, ITU-T 0.181 specific mapping path signal label is selected.

Example(s)

* SENS:DATA:TEL:SON:HOP:PATH:LAB:EXP EQU
* SENS:DATA:TEL:SON:HOP:PATH:LAB:EXP?
Returns EQUIPPED

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

See Also

* SENSE[1..n]:DATA:TELEcom:SONet:HOP:PATH
LABel:EXPEcted

**:FETCh[1..n]:DATA:TELEcom:SONet:HOP:PATH:
LABel?**

Description	<p>This query returns the expected path signal label of High Order Path (HOP).</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SONet:HOP:PATH: LABel?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Label></p>
Response(s)	<p>Label: The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element. Returns the path signal label for the receiver.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:HOP:PATH:LAB EQU * FETC:DATA:TEL:SON:HOP:PATH:LAB? Returns the path signal label.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:HOP: PATH:LABel</p>

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

Description	<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, this value is set to OFF.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SONet:HOP:PUNeq <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Signal Label Mismatch for the expected message as well as UNEQ-P (Unequipped - Path) monitoring.</p>
Example(s)	<pre>* SENS:DATA:TEL:SON:HOP:PUN ON * SENS:DATA:TEL:SON:HOP:PUN? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:SONet:HOP: PUNeq?</pre>

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

Description	This query returns the status of Signal Label Mismatch for the expected message as well as Unequipped - Path (UNEQ-P) monitoring. At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:SONet:HOP: PUNeq?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of Payload Label Mismatch - Path (PLM-P) / Unequipped - Path (UNEQ-P).
Example(s)	* SENS:DATA:TEL:SON:HOP:PUN ON * SENS:DATA:TEL:SON:HOP:PUN? Returns 1
See Also	* SENSe[1..n]:DATA:TELEcom:SONet:HOP: PUNeq

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:TYPE****Description**

This command selects the type of High Order Path (HOP) error for automated injection.

At *RST, this value is set to BERRor.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:AUTomated:TYPE<wsp>BERRor|REI

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROR:
HOP:PATH:AUTomated:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the type of High Order Path (HOP) error for automated injection.</p> <p>BERRor, selects the type of High Order Path (HOP) error as B3.</p> <p>REI, selects the REI-P (Remote Error Indicator - Path) error when bits 1 through 4 of the G1 byte contain one pattern from the following binary range: "#B0001" through "#B1000" (1 to 8).</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERROR:HOP:PATH:AUTomated:TYPE?</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERROR:HOP:PATH:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERROR:HOP:PATH:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:TYPE?**

Description	This query returns the type of High Order Path (HOP) error for automated injection. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELEcom:SONet:ERROR: HOP:PATH:AUTomated:TYPE?

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of High Order Path (HOP) error for the automated injection.</p> <p>BERROR, 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>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERROR:HOP:PATH:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERROR:HOP:PATH:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERROR:HOP:PATH:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:RATE****Description**

This command sets the injection rate for the selected High Order Path (HOP) error.

At *RST, this value is device dependent.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:RATE <wsp> <Rate>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:RATE**

Parameter(s) Rate:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected High Order Path (HOP) error.

Example(s) * SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:
TYPE BERR
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:
RATE 1.0E-10
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:
RATE? Returns 1.0E-10

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
HOP:PATH:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: HOP:PATH:AUTomated:RATE?[<wsp> MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injected rate will be returned.</p>
Response Syntax	<pre><Rate></pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:RATE?**

Response(s)	Rate: The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the injection rate for the selected High Order Path (HOP) error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT: TYPE BERR * SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT: RATE 1.0E-10 * SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT: RATE? Returns 1.0E-10
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated****Description**

This command enables or disables the selected automated High Order Path (HOP) error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated<wsp><Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated**

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated High Order Path (HOP) error injection.

Example(s) * SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:
TYPE BERR
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:
RATE 1.0E-10
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT ON
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT
Returns 1

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated?**

Description	This query returns the status of automated High Order Path (HOP) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated?**

Response(s)

Set:
The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the status of automated High Order Path (HOP) error injection.

Example(s)

* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:
TYPE BERR
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:
RATE 1.0E-10
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT ON
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:CONTInuous****Description**

This command enables or disables the continuous rate of automated High Order Path (HOP) error injection.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:CONTInuous<wsp>
<Set>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:CONTInuous

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated High Order Path (HOP) error injection continuously.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:TYPE BERR* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:CONT ON* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:CONT? Returns 1* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT ON
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:AUTomated* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:CONTInuous?**

Description	This query returns status of the continuous rate of automated High Order Path (HOP) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:CONTInuous?**

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of continuous rate of automated High Order Path (HOP) error injection.

Example(s)

* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:
TYPE BERR

* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:
CONT ON

* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT:
CONT? Returns 1

* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AUT ON

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AUTomated:CONTInuous

:SOURce[1..n]:DATA:TELEcom:SONet:LOP: PATH:LABel

Description	<p>This command sets the path signal label (V5) of Low Order Path (LOP) for the transmitter.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:LOP: PATH:LABel<wsp>UNEQuipped EQUipped ASYNchronous BISYNch BYSYNch SIGNal TEST VTais</pre>
Parameter(s)	<p>Label:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>UNEQuipped EQUipped ASYNchronous BISYNch BYSYNch SIGNal TEST VTais.</p> <p>Selects the path signal label.</p> <p>UNEQuipped, selects the Unequipped path signal label.</p> <p>EQUipped, selects the Equipped non-specific path signal label.</p> <p>ASYNchronous, selects the Asynchronous path signal label.</p> <p>BISYNch, selects the Bit Synchronous path signal label.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:LOP:
PATH:LABel**

BYSynch, selects the Byte Synchronous path signal label.

SIGNAL, selects the extended path Signal.

TEST, selects the Test signal, ITU-T 0.181 specific mapping path signal label.

VTais, selects the VT (Virtual Tributary) SPE (Synchronous Payload Envelope) AIS (Alarm Indication Signal)(TCM).

Example(s)

* SOUR:DATA:TEL:SON:LOP:PATH:LAB EQU

* SOUR:DATA:TEL:SON:LOP:PATH:LAB?

Returns EQUIPPED

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:LOP:PATH:LABel?

**:SOURce[1..n]:DATA:TELEcom:SONet:LOP:
PATH:LABel?**

Description	<p>This query returns the path signal label (V5) of Low Order Path (LOP) for the transmitter.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:LOP: PATH:LABel?
Parameter(s)	None
Response Syntax	<Label>
Response(s)	<p>Label:</p> <p>The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element. Returns the path signal label.</p> <p>UNEQUIPPED, Unequipped path signal label is selected.</p> <p>EQUIPPED, Equipped non-specific path signal label is selected.</p> <p>ASYNCHRONOUS, Asynchronous path signal label is selected.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:LOP:
PATH:LABel?**

BISYNCH, Bit Synchronous path signal label is selected.

BYSYNCH, Byte Synchronous path signal label is selected.

SIGNAL, Extended path Signal is selected.

TEST, Test signal, ITU-T 0.181 specific mapping path signal label is selected.

VT AIS, Virtual Tributary (VT) Synchronous Payload Envelope (SPE) Alarm Indication Signal (AIS)(TCM) is selected.

Example(s)

* SOUR:DATA:TEL:SON:LOP:PATH:LAB EQU

* SOUR:DATA:TEL:SON:LOP:PATH:LAB?

Returns EQUIPPED

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:LOP:
PATH:LABel

:SENSe[1..n]:DATA:TELEcom:SONet:LOP:PATH: LABel:EXPEcted

Description	This command sets the expected path signal label (V5) of Low Order Path (LOP) for the receiver. At *RST, this value is device dependent.
Syntax	:SENSe[1..n]:DATA:TELEcom:SONet:LOP:PATH: LABel:EXPEcted<wsp>EQUipped ASYNchronous BISYNch BYSYnch SIGNal TEST
Parameter(s)	<p>Label:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: EQUipped ASYNchronous BISYNch BYSYnch SIGNal TEST.</p> <p>Selects the path signal label.</p> <p>EQUipped, selects the Equipped non-specific path signal label.</p> <p>ASYNchronous, selects the Asynchronous path signal label.</p> <p>BISYNch, selects the Bit Synchronous path signal label.</p>

:SENSe[1..n]:DATA:TELEcom:SONet:LOP:PATH: LABel:EXPEcted

BYSYnch, selects the Byte Synchronous path signal label.

SIGNal, selects the extended path Signal.

TEST, selects the Test signal, ITU-T 0.181 specific mapping path signal label.

Example(s)

* SENS:DATA:TEL:SON:LOP:PATH:LAB:EXP EQU

* SENS:DATA:TEL:SON:LOP:PATH:LAB:EXP?

Returns EQUIPPED

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SENSe[1..n]:DATA:TELEcom:SONet:LOP:PATH:
LABel:EXPEcted?

**:SENSE[1..n]:DATA:TELEcom:SONet:LOP:PATH:
LABel:EXPEcted?**

Description	<p>This query returns the expected path signal label (V5) of Low Order Path (LOP) for the receiver.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	:SENSE[1..n]:DATA:TELEcom:SONet:LOP:PATH: LABel:EXPEcted?
Parameter(s)	None
Response Syntax	<Label>
Response(s)	<p>Label:</p> <p>The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element. Returns the path signal label.</p> <p>EQUIPPED, Equipped non-specific path signal label is selected.</p> <p>ASYNCHRONOUS, Asynchronous path signal label is selected.</p> <p>BISYNCH, Bit Synchronous path signal label is selected.</p>

:SENSe[1..n]:DATA:TELEcom:SONet:LOP:PATH: LABel:EXPEcted?

BYSYNCH, Byte Synchronous path signal label is selected.

SIGNAL, Extended path Signal is selected.

TEST, Test signal, ITU-T 0.181 specific mapping path signal label is selected.

Example(s)

* SENS:DATA:TEL:SON:LOP:PATH:LAB:EXP EQU

* SENS:DATA:TEL:SON:LOP:PATH:LAB:EXP?

Returns EQUIPPED

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SENSe[1..n]:DATA:TELEcom:SONet:LOP:PATH:
LABel:EXPEcted

**:FETCh[1..n]:DATA:TELEcom:SONet:LOP:PATH:
LABel?**

Description	<p>This query returns the expected path signal label of Low Order Path (LOP) for the receiver.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SONet:LOP:PATH: LABel?
Parameter(s)	None
Response Syntax	<Label>
Response(s)	<p>Label:</p> <p>The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element. Returns the path signal label for the receiver.</p>

:FETCh[1..n]:DATA:TELEcom:SONet:LOP:PATH: LABel?

Example(s)	* SOUR:DATA:TEL:SON:LOP:PATH:LAB EQU * FETC:DATA:TEL:SON:LOP:PATH:LAB? Returns the path signal label.
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:LOP:PATH:LABel

**:SENSe[1..n]:DATA:TELEcom:SONet:LOP:
PUNeq**

Description	<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, this value is set to OFF.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SONet:LOP:PUNeq <wsp><Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Signal Label Mismatch for the expected message as well as UNEQ-P (Unequipped - Path) monitoring.</p>

**:SENSe[1..n]:DATA:TELEcom:SONet:LOP:
PUNeq**

Example(s) * SENS:DATA:TEL:SON:LOP:PUN ON
 * SENS:DATA:TEL:SON:LOP:PUN? Returns 1

Note FTB/IQS-8140 Transport Blazer does not support
 this command.

See Also * SENSe[1..n]:DATA:TELEcom:SONet:LOP:
 PUNeq?

**:SENSe[1..n]:DATA:TELEcom:SONet:LOP:
PUNeq?**

Description	<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, this value is set to OFF.</p>
Syntax	<code>:SENSe[1..n]:DATA:TELEcom:SONet:LOP: PUNeq?</code>
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Payload Label Mismatch - VT (PLM-V)/Unequipped - VT (UNEQ-V).</p>

:SENSe[1..n]:DATA:TELEcom:SONet:LOP: PUNeq?

Example(s)	* SENS:DATA:TEL:SON:LOP:PUN ON * SENS:DATA:TEL:SON:LOP:PUN? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SENSe[1..n]:DATA:TELEcom:SONet:LOP:PUNeq

:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LOP:PATH:AUTomated:TYPE

Description	<p>This command selects the type of Low Order Path (LOP) error for automated injection.</p> <p>At *RST, this value is set to BIP2.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LOP:PATH:AUTomated:TYPE<wsp>BIP2 REI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 REI.</p> <p>Selects the type of Low Order Path (LOP) error for automated injection.</p> <p>BIP2, selects the BIP-2 (Bit-Interleaved Parity - 2 bits) error which indicates a parity error by performing a routine even-parity check over all VT1.5 bytes of the previous frame of a composite signal (VT1.5/VT2/VT6).</p> <p>REI, selects the REI-V (Remote Error Indication - VT) error when bit 3 of the V5 byte is set to "1".</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:TYPE

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:TYPE BIP2* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:TYPE? Returns BIP2
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:TYPE?* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LOP:PATH:AUTomated:TYPE?**

Description	<p>This query returns the type of Low Order Path (LOP) error for automated injection.</p> <p>At *RST, this value is set to BIP2.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LOP:PATH:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of Low Order Path (LOP) error for the automated injection.</p> <p>BIP2, Bit-Interleaved Parity - 2 bits (BIP2) is selected as Low Order Path (LOP) error.</p> <p>REI, Remote Error Indication - VT (REIV) is selected as Low Order Path (LOP) error.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:TYPE?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:TYPE BIP2* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:TYPE? Returns BIP2
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated:RATE**

Description	<p>This command sets the injection rate for the selected Low Order Path (LOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:RATE<wsp><Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the injection rate for the selected Low Order Path (LOP) error.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:RATE

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:TYPE BIP2* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:RATE 1.0E-10* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:RATE? Returns 1.0E-10
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:RATE?* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LOP:PATH:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Low Order Path (LOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LOP:PATH:AUTomated:RATE?[<wsp> MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injection rate will be returned.</p>
Response Syntax	<pre><Rate></pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated:RATE?**

Response(s)	<p>Rate: The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the injection rate for the selected High Order Path (HOP) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:TYPE BIP2 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:RATE 1.0E-10 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:RATE? Returns 1.0E-10</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated**

Description	<p>This command enables or disables the selected automated Low Order Path (LOP) error at the rate specified or continuously.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated Low Order Path (LOP) error injection.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated

Example(s)

- * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:TYPE BIP2
- * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:RATE 1.0E-10
- * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT ON
- * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT?

Returns 1

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:TYPE
- * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:RATE
- * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated?**

Description	<p>This query returns the status of automated Low Order Path (LOP) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated?</code>
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of automated Low Order Path (LOP) error injection.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated?

Example(s) * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:
TYPE BIP2
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:
RATE 1.0E-10
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT ON
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT?
Returns 1

Note FTB/IQS-8140 Transport Blazer does not support
this query.

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated:CONTInuous**

Description	<p>This command enables or disables the continuous rate of automated Low Order Path (LOP) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:CONTInuous <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated Low Order Path (LOP) error injection continuously.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:CONTInuous

Example(s)

- * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:TYPE BIP2
- * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:CONT ON
- * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:CONT? Returns 1
- * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT ON

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:TYPE
- * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated
- * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated:CONTInuous?**

Description	<p>This query returns the status of continuous rate of automated Low Order Path (LOP) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AUTomated:CONTInuous?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuous rate of automated Low Order Path (LOP) error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated:CONTInuous?**

Example(s) * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:
TYPE BIP2
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:
CONT ON
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT:
CONT? Returns 1
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AUT ON

Note FTB/IQS-8140 Transport Blazer does not support
this query.

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AUTomated:CONTInuous

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion:TYPE

Description	<p>This command selects the type of section alarm.</p> <p>At *RST, this value is set to LOF1.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion:TYPE<wsp>LOF1 SEF1
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF1 SEF1.</p> <p>Selects the type of section alarm.</p> <p>LOF1, selects the non-valid framing bytes as LOF (Loss of Frame).</p> <p>SEF1, selects the four consecutive error framing patterns as SEF (Severely Errored Framing).</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:SECT:TYPE LOF1</p> <p>* SOUR:DATA:TEL:SON:ALAR:SECT:TYPE?</p> <p>Returns LOF1</p>
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion:TYPE?

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:TYPE?

Description	This query returns the type of section alarm. At *RST, this value is set to LOF1.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	Alarm: The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of section alarm. LOF1, Loss of Frame (LOF) is selected as section alarm.

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:TYPE?**

SEF1, Severely Errored Framing (SEF) is selected as section alarm.

Example(s)

* SOUR:DATA:TEL:SON:ALAR:SECT:TYPE LOF1
* SOUR:DATA:TEL:SON:ALAR:SECT:TYPE?
Returns LOF1

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:TYPE

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION**

Description	<p>This command enables or disables the section alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION<wsp><Set></p>
Parameter(s)	<p>Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the section alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:SECT:TYPE LOF1* SOUR:DATA:TEL:SON:ALAR:SECT ON* SOUR:DATA:TEL:SON:ALAR:SECT? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTION:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTION?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTion?**

Description	This query returns the status of section alarm. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of section alarm generation.
Example(s)	* SOUR:DATA:TEL:SON:ALAR:SECT:TYPE LOF1 * SOUR:DATA:TEL:SON:ALAR:SECT ON * SOUR:DATA:TEL:SON:ALAR:SECT? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:MANual:TYPE**

Description This command sets the manual type of section error.

At *RST, this value is set to BERRor.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:MANual:TYPE <wsp> BERRor|FAS

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor|FAS.
Selects the type of Section error.
BERRor, selects the type of section error as B1.
FAS, selects the type of section error as Frame Alignment Signal (FAS).

Example(s) * SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE
BERR
* SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE?
Returns BERROR

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:MANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:MANual:TYPE?**

Description	<p>This query returns the manual type of section error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of section error. BERROR, B1 is selected as section error. FAS, Frame Alignment Signal (FAS) is selected as section error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE? Returns BERROR</p>
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROR:
SECTION:AMOUNT**

Description This command sets the amount of section error to be injected into the instrument.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERROR:
SECTION:AMOUNT <wsp> <Amount>
|MAXimum|MINimum

Parameter(s) Amount:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AMOUNT**

Sets the amount of section error to be injected.
Choices are 1 through 50.

Example(s)

```
* SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE  
BERR  
* SOUR:DATA:TEL:SON:ERR:SECT:AMO 15  
* SOUR:DATA:TEL:SON:ERR:SECT:AMO?  
Returns 15
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:  
SECTION:MANual:TYPE  
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:  
SECTION:AMOUNT?
```

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AMOUnt?**

Description	<p>This query returns the amount of section error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:AMOUnt?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The response data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<p><Amount></p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AMOut?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of section error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE BERR * SOUR:DATA:TEL:SON:ERR:SECT:AMO 15 * SOUR:DATA:TEL:SON:ERR:SECT:AMO? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:AMOut

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:INJect

Description	<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>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:INJect
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE BERR* SOUR:DATA:TEL:SON:ERR:SECT:AMO 15* SOUR:DATA:TEL:SON:ERR:SECT:INJ
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:AMOUNT

:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:HISTory?

Description	<p>This query returns the history status of section alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:HISTory? <wsp>LOF1 SEF1 TIMS
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF1 SEF1 TIMS.</p> <p>Selects the type of section alarm.</p> <p>LOF1, selects the non-valid framing bytes as LOF (Loss of Frame).</p> <p>SEF1, selects the four consecutive error framing patterns as SEF (Severely Errored Framing).</p> <p>TIMS, selects the sampled section trace strings match the expected message value as TIM-S (Trace Identifier Mismatch - Section).</p>
Response Syntax	<History>

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: SECTION:HISTory?

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:SECT:TYPE LOF1* SOUR:DATA:TEL:SON:ALAR:SECT ON* FETC:DATA:TEL:SON:ALAR:SECT:HIST? LOF1 <p>Returns the alarm history.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELecom:SONet:ALARm:SECTION:MANual:TYPE* SOURce[1..n]:DATA:TELecom:SONet:ALARm:SECTION

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
SECTIon:SECOnds?**

Description	<p>This query returns the number of seconds within which section alarm occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:SECOnds?<wsp>LOF1 SEF1 TIMS</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF1 SEF1 TIMS.</p> <p>Selects the type of section alarm.</p> <p>LOF1, selects the non-valid framing bytes as LOF (Loss of Frame).</p> <p>SEF1, selects the four consecutive error framing patterns as SEF (Severely Errored Framing).</p> <p>TIMS, selects the sampled section trace strings match the expected message value as TIM-S (Trace Identifier Mismatch - Section).</p>
Response Syntax	<pre><Seconds></pre>

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: SECTion:SEConds?

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of section alarm.
Example(s)	* SOUR:DATA:TEL:SON:ALAR:SECT:TYPE LOF1 * SOUR:DATA:TEL:SON:ALAR:SECT ON * FETC:DATA:TEL:SON:ALAR:SECT:SEC? LOF1 Returns the number of seconds of section alarm.
See Also	* SOURce[1..n]:DATA:TELecom:SONet:ALARm: SECTion:MANual:TYPE * SOURce[1..n]:DATA:TELecom:SONet:ALARm: SECTion

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
SECTIon:CURRent?**

Description	<p>This query returns the current status of section alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:CURRent?<wsp>LOF1 SEF1 TIMS</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF1 SEF1 TIMS.</p> <p>Selects the type of section alarm.</p> <p>LOF1, selects the non-valid framing bytes as LOF (Loss of Frame).</p> <p>SEF1, selects the four consecutive error framing patterns as SEF (Severely Errored Framing).</p> <p>TIMS, selects the sampled section trace strings match the expected message value as TIM-S (Trace Identifier Mismatch - Section).</p>
Response Syntax	<pre><Current></pre>

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: SECTion:CURRent?

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:SECT:TYPE LOF1* SOUR:DATA:TEL:SON:ALAR:SECT ON* FETC:DATA:TEL:SON:ALAR:SECT:CURR? LOF1 <p>Returns the current alarm status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELecom:SONet:ALARm:SECTion:MANual:TYPE* SOURce[1..n]:DATA:TELecom:SONet:ALARm:SECTion

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
SECTIon:HISTory?**

Description	<p>This query returns the history status of section error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:HISTory? <wsp>BERRor FAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS.</p> <p>Selects the type of section error.</p> <p>BERRor, selects the type of section error as B1.</p> <p>FAS, selects the type of section error as Frame Alignment Signal (FAS).</p>
Response Syntax	<pre><History></pre>

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:HISTory?

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE BERR* SOUR:DATA:TEL:SON:ERR:SECT:AMO 15* SOUR:DATA:TEL:SON:ERR:SECT:INJ* FETC:DATA:TEL:SON:ERR:SECT:HIST? BERR <p>Returns the current error history.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:SECTION:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:SECTION:AMOUNT* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:SECTION:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERROR:
SECTION:SEConds?**

Description	<p>This query returns the number of seconds within which section error occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ERROR: SECTION:SEConds?<wsp>BERRor FAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS.</p> <p>Selects the type of section error.</p> <p>BERRor, selects the type of section error as B1.</p> <p>FAS, selects the type of section error as Frame Alignment Signal (FAS).</p>
Response Syntax	<pre><Seconds></pre>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of section error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE BERR * SOUR:DATA:TEL:SON:ERR:SECT:AMO 15 * SOUR:DATA:TEL:SON:ERR:SECT:INJ * FETC:DATA:TEL:SON:ERR:SECT:SEC? BERR Returns the number of errored seconds.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:AMOUnt * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERROr:
SECTIon:CURREnt?**

Description	<p>This query returns the current status of section error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ERROr: SECTIon:CURREnt?<wsp>BERRor FAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS.</p> <p>Selects the type of section error.</p> <p>BERRor, selects the type of section error as B1.</p> <p>FAS, selects the type of section error as Frame Alignment Signal (FAS).</p>
Response Syntax	<pre><Current></pre>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:CURRent?**

Response(s) Current:
The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the current status of section 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:SON:ERR:SECT:MAN:TYPE
BERR
* SOUR:DATA:TEL:SON:ERR:SECT:AMO 15
* SOUR:DATA:TEL:SON:ERR:SECT:INJ
* FETC:DATA:TEL:SON:ERR:SECT:CURR? BERR
Returns the current error status.

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:AMOUnt
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
SECTIon:COUNT?**

Description	<p>This query returns the count of section error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:COUNT? <wsp>BERRor FAS</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS.</p> <p>Selects the type of section error.</p> <p>BERRor, selects the type of section error as B1.</p> <p>FAS, selects the type of section error as Frame Alignment Signal (FAS).</p>
Response Syntax	<code><Count></code>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:COUNT?**

Response(s)	Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of section error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE BERR * SOUR:DATA:TEL:SON:ERR:SECT:AMO 15 * SOUR:DATA:TEL:SON:ERR:SECT:INJ * FETC:DATA:TEL:SON:ERR:SECT:COUN? BERR Returns the error count.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:AMOUNT * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
SECTIon:RATE?**

Description	<p>This query returns the current rate of section error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:RATE? <wsp>BERRor FAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS.</p> <p>Selects the type of section error.</p> <p>BERRor, selects the type of section error as B1.</p> <p>FAS, selects the type of section error as Frame Alignment Signal (FAS).</p>
Response Syntax	<pre><Rate></pre>

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of section error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:SECT:MAN:TYPE BERR* SOUR:DATA:TEL:SON:ERR:SECT:AMO 15* SOUR:DATA:TEL:SON:ERR:SECT:INJ* FETC:DATA:TEL:SON:ERR:SECT:RATE? BERR <p>Returns the error rate.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:SECTion:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:SECTion:AMOUnt* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:SECTion:INJect

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:AUTomated:TYPE**

Description This command selects the type of Section error for automated injection.

At *RST, this value is set to BERRor.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:AUTomated:TYPE<wsp>BERRor

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:AUTomated:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter is: BERRor.</p> <p>Selects the type of Section error for automated injection.</p> <p>BERRor, selects the type of section error as B1.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:SECT:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:AUT:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:TYPE?</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTIon:AUTomated:TYPE?**

Description	This query returns the type of Section error for automated injection. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:TYPE?

Response(s)	<p>Error</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of Section error for the automated injection.</p> <p>BERROR, B1 is selected as section error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:SECT:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:AUT:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AUTomated:RATE**

Description	<p>This command sets the injection rate for the selected Section error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:AUTomated:RATE<wsp> <Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AUTomated:RATE**

Sets the injection rate for the selected Section error.

Example(s)

```
* SOUR:DATA:TEL:SON:ERR:SECT:AUT:TYPE  
BERR  
* SOUR:DATA:TEL:SON:ERR:SECT:AUT:RATE  
1.0E-10  
* SOUR:DATA:TEL:SON:ERR:SECT:AUT:RATE?  
Returns 1.0E-10
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:  
SECTION:AUTomated:TYPE  
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:  
SECTION:AUTomated:RATE?  
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:  
SECTION:AUTomated
```


**:SOURce[1..n]:DATA:TELEcom:SONet:ERROR:
SECTIon:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Section error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERROR: SECTIon:AUTomated:RATE? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injection rate will be returned.</p>
Response Syntax	<pre><Rate></pre>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Section error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:SECT:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:AUT:RATE 1.0E-10</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:AUT:RATE? Returns 1.0E-10</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:AUTomated****Description**

This command enables or disables the selected automated Section error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:AUTomated <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:AUTomated**

Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated Section error injection.
Example(s)	* SOUR:DATA:TEL:SON:ERR:SECT:AUT:TYPE BERR * SOUR:DATA:TEL:SON:ERR:SECT:AUT:RATE 1.0E-10 * SOUR:DATA:TEL:SON:ERR:SECT:AUT ON * SOUR:DATA:TEL:SON:ERR:SECT:AUT Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTIon:AUTomated?**

Description	This query returns the status of automated section error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:AUTomated?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated Section error injection.
Example(s)	* SOUR:DATA:TEL:SON:ERR:SECT:AUT:TYPE BERR * SOUR:DATA:TEL:SON:ERR:SECT:AUT:RATE 1.0E-10 * SOUR:DATA:TEL:SON:ERR:SECT:AUT ON * SOUR:DATA:TEL:SON:ERR:SECT:AUT Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTIon:AUTomated:CONTInuous****Description**

This command enables or disables the continuous rate of automated Section error injection.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTIon:AUTomated:CONTInuous <wsp>
<Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AUTomated:CONTInuous**

Parameter(s)

Set:

The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated Section error injection continuously.

Example(s)

* SOUR:DATA:TEL:SON:ERR:SECT:AUT:TYPE
BERR
* SOUR:DATA:TEL:SON:ERR:SECT:AUT:CONT ON
* SOUR:DATA:TEL:SON:ERR:SECT:AUT:CONT?
Returns 1
* SOUR:DATA:TEL:SON:ERR:SECT:AUT ON

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AUTomated
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTIon:AUTomated:CONTInuous?**

Description	This query returns the status of continuous rate of automated Section error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTIon:AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AUTomated:CONTInuous?**

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of continuous rate of automated Section error injection.

Example(s)

* SOUR:DATA:TEL:SON:ERR:SECT:AUT:TYPE
BERR

* SOUR:DATA:TEL:SON:ERR:SECT:AUT:CONT ON

* SOUR:DATA:TEL:SON:ERR:SECT:AUT:CONT?

Returns 1

* SOUR:DATA:TEL:SON:ERR:SECT:AUT ON

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AUTomated

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:AUTomated:CONTInuous

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:TYPE**

Description	<p>This command selects the type of line alarm.</p> <p>At *RST, this value is set to AIS.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:TYPE<wsp>AIS RDI</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS RDI.</p> <p>Selects the type of line alarm.</p> <p>AIS, selects the AIS-L (Alarm Indication Signal - Line) which generates a "111" pattern for the bits 6, 7 and 8 of the K2 byte.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:TYPE**

RDI, selects 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:SON:ALAR:LINE:TYPE AIS
* SOUR:DATA:TEL:SON:ALAR:LINE:TYPE?
Returns AIS

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:TYPE?**

Description	<p>This query returns the type of line alarm.</p> <p>At *RST, this value is set to AIS.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. 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>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:LINE:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ALAR:LINE:TYPE?</p> <p>Returns AIS</p>
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:TYPE

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE**

Description	<p>This command enables or disables the line alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE<wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables and disables the line alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LINE:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:LINE ON* SOUR:DATA:TEL:SON:ALAR:LINE? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LINE:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LINE?

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE?

Description	<p>This query returns the status of line alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of line alarm generation.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:LINE:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ALAR:LINE ON</p> <p>* SOUR:DATA:TEL:SON:ALAR:LINE? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:MANual:TYPE**

Description This command sets the manual type of line error.

At *RST, this value is set to BERRor.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:MANual:TYPE<wsp>BERRor|REI

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor|REI.
Selects the type of Line error.
BERRor, selects the type of line error as B2.
REI, selects the type of line error as REI (Remote Error Indication - Line).

Example(s) * SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE
BERR
* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE?
Returns BERROR

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:MANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:MANual:TYPE?**

Description	<p>This query returns the manual type of line error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:MANual:TYPE?
Response Syntax	<Error>
Parameter(s)	None
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of line error.</p> <p>BERROR, B2 is selected as line error.</p> <p>REI, Remote Error Indication - Line (REI-L) is selected as line error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE? Returns BERROR</p>
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:MANual:TYPE

:SOURce[1..n]:DATA:TELEcom:SONet:ERROR: LINE:AMOUNT

Description	<p>This command sets the amount of line error to be injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERROR: LINE:AMOUNT<wsp><Amount> MAXimum MINimum</pre>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AMOUNT**

Sets the amount of line error to be injected.
Choices are 1 through 50.

Example(s)

```
* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE  
BERR  
* SOUR:DATA:TEL:SON:ERR:LINE:AMO 15  
* SOUR:DATA:TEL:SON:ERR:LINE:AMO?  
Returns 15
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:  
LINE:MANual:TYPE  
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:  
LINE:AMOUNT?
```

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LINE:AMOUnt?**

Description	<p>This query returns the amount of line error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LINE:AMOUnt?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<p><Amount></p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AMOut?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of line error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:AMO 15 * SOUR:DATA:TEL:SON:ERR:LINE:AMO? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AMOut

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:INJect

Description	<p>This command injects the type of line error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:INJect
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE BERR* SOUR:DATA:TEL:SON:ERR:LINE:AMO 15* SOUR:DATA:TEL:SON:ERR:LINE:INJ
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AMOUNT

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:HISTory?**

Description	<p>This query returns the history status of line alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LINE:HISTory? <wsp> AIS RDI</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS RDI.</p> <p>Selects the type of line alarm.</p> <p>AIS, selects 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, selects the RDI-L (Remote Defect Indication - Line) which generates a "110" pattern for the bits 6, 7 and 8 of the K2 byte.</p>
Response Syntax	<pre><History></pre>

:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LINE:HISTory?

Response(s)

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the history status of line 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)

* SOUR:DATA:TEL:SON:ALAR:LINE:TYPE AIS

* SOUR:DATA:TEL:SON:ALAR:LINE ON

* FETC:DATA:TEL:SON:ALAR:LINE:HIST? AIS

Returns the alarm history.

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:TYPE

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:SECOnds?**

Description	<p>This query returns the number of seconds within which line alarm occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LINE:SECOnds? <wsp>AIS RDI</pre>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS RDI. Selects the type of line alarm. AIS, selects the AIS-L (Alarm Indication Signal - Line) which generates a "111" pattern for the bits 6, 7 and 8 of the K2 byte. RDI, selects the RDI-L (Remote Defect Indication - Line) which generates a "110" pattern for the bits 6, 7 and 8 of the K2 byte.</p>
Response Syntax	<pre><Seconds></pre>

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of line alarm.
Example(s)	* SOUR:DATA:TEL:SON:ALAR:LINE:TYPE AIS * SOUR:DATA:TEL:SON:ALAR:LINE ON * FETC:DATA:TEL:SON:ALAR:LINE:SEC? AIS Returns the number of ALARed seconds.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LINE:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LINE

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:CURRent?**

Description	<p>This query returns the current status of line alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LINE:CURRent? <wsp>AIS RDI</pre>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS RDI. Selects the type of line alarm. AIS, selects the AIS-L (Alarm Indication Signal - Line) which generates a "111" pattern for the bits 6, 7 and 8 of the K2 byte. RDI, selects the RDI-L (Remote Defect Indication - Line) which generates a "110" pattern for the bits 6, 7 and 8 of the K2 byte.</p>
Response Syntax	<pre><Current></pre>

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LINE:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:LINE ON* FETC:DATA:TEL:SON:ALAR:LINE:CURR? AIS <p>Returns the current alarm status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LINE:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LINE

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:HISTory?**

Description	<p>This query returns the history status of line error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LINE: HISTory? <wsp>BERRor REI</p>
Parameter(s)	<p>Error: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI. Selects the type of line error. BERRor, selects the type of line error as B2. REI, selects the type of line error as REI-L (Remote Defect Indication Line).</p>
Response Syntax	<p><History></p>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE BERR* SOUR:DATA:TEL:SON:ERR:LINE:AMO 15* SOUR:DATA:TEL:SON:ERR:LINE:INJ* FETC:DATA:TEL:SON:ERR:LINE:HIST? BERR <p>Returns the error history status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LINE:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LINE:AMOUNT* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LINE:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:SEConds?**

Description	<p>This query returns the number of seconds within which line error occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LINE:SEConds?<wsp>BERRor REI</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the type of line error.</p> <p>BERRor, selects the type of line error as B2.</p> <p>REI, selects the type of line error as REI-L (Remote Defect Indication Line).</p>
Response Syntax	<code><Seconds></code>

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: LINE:SEConds?

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of errored seconds.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:AMO 15 * SOUR:DATA:TEL:SON:ERR:LINE:INJ * FETC:DATA:TEL:SON:ERR:LINE:SEC? BERR Returns the number of errored seconds.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AMOUNT * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:CURRent?**

Description	<p>This query returns the current status of line error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LINE: CURRent? <wsp>BERRor REI</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the type of line error.</p> <p>BERRor, selects the type of line error as B2.</p> <p>REI, selects the type of line error as REI-L (Remote Defect Indication Line).</p>
Response Syntax	<code><Current></code>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE BERR* SOUR:DATA:TEL:SON:ERR:LINE:AMO 15* SOUR:DATA:TEL:SON:ERR:LINE:INJ* FETC:DATA:TEL:SON:ERR:LINE:CURR? BERR <p>Returns the current error status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LINE:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LINE:AMOUNT* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LINE:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:COUNT?**

Description	<p>This query returns the count of line error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LINE: COUNT?<wsp>BERRor REI</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the type of line error.</p> <p>BERRor, selects the type of line error as B2.</p> <p>REI, selects the type of line error as REI-L (Remote Defect Indication Line).</p>
Response Syntax	<code><Count></code>

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: LINE:COUNT?

Response(s)	Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of line error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:AMO 15 * SOUR:DATA:TEL:SON:ERR:LINE:INJ * FETC:DATA:TEL:SON:ERR:LINE:COUN? BERR Returns the error count.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AMount * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:RATE?**

Description	<p>This query returns the current rate of line error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LINE: RATE? <wsp>BERRor REI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the type of line error.</p> <p>BERRor, selects the type of line error as B2.</p> <p>REI, selects the type of line error as REI-L (Remote Defect Indication Line).</p>
Response Syntax	<pre><Rate></pre>

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: LINE:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of line error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LINE:MAN:TYPE BERR* SOUR:DATA:TEL:SON:ERR:LINE:AMO 15* SOUR:DATA:TEL:SON:ERR:LINE:INJ* FETC:DATA:TEL:SON:ERR:LINE:RATE? BERR <p>Returns the error rate.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LINE:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LINE:AMOUNT* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LINE:INJect

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:TYPE**

Description	<p>This command selects the type of Line error for automated injection.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:TYPE<wsp>BERRor REI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the type of Line error for automated injection.</p> <p>BERRor, selects the type of line error as B2.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:TYPE**

REI, selects the type of line error as REIL
(Remote Error Indication - Line).

Example(s)

* SOUR:DATA:TEL:SON:ERR:LINE:AUT:TYPE
BERR
* SOUR:DATA:TEL:SON:ERR:LINE:AUT:TYPE?
Returns BERROR

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:TYPE?
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LINE:AUTomated:TYPE?**

Description	This query returns the type of Line error for automated injection. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LINE:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of Line error for the automated injection.</p> <p>BERROR, B2 is selected as line error.</p> <p>REI, Remote Error Indication - Line (REI-L) is selected as line error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:LINE:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:LINE:AUT:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:RATE**

Description	<p>This command sets the injection rate for the selected Line error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:RATE<wsp><Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:RATE**

Sets the injection rate for the selected Line error.

Example(s)

```
* SOUR:DATA:TEL:SON:ERR:LINE:AUT:TYPE  
BERR  
* SOUR:DATA:TEL:SON:ERR:LINE:AUT:RATE  
1.0E-10  
* SOUR:DATA:TEL:SON:ERR:LINE:AUT:RATE?  
Returns 1.0E-10
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:  
LINE:AUTomated:TYPE  
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:  
LINE:AUTomated:RATE?  
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:  
LINE:AUTomated
```

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LINE:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Line error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LINE:AUTomated:RATE?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injection rate will be returned.</p>
Response Syntax	<pre><Rate></pre>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Line error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:LINE:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:LINE:AUT:RATE 1.0E-10</p> <p>* SOUR:DATA:TEL:SON:ERR:LINE:AUT:RATE? Returns 1.0E-10</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated****Description**

This command enables or disables the selected automated line error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated**

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated line error injection.

Example(s) * SOUR:DATA:TEL:SON:ERR:LINE:AUT:TYPE
BERR
* SOUR:DATA:TEL:SON:ERR:LINE:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:SON:ERR:LINE:AUT ON
* SOUR:DATA:TEL:SON:ERR:LINE:AUT?
Returns 1

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated?**

Description	This query returns the status of automated line error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of automated line error injection.

Example(s)

* SOUR:DATA:TEL:SON:ERR:LINE:AUT:TYPE
BERR

* SOUR:DATA:TEL:SON:ERR:LINE:AUT:RATE
1.0E-10

* SOUR:DATA:TEL:SON:ERR:LINE:AUT ON

* SOUR:DATA:TEL:SON:ERR:LINE:AUT?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:CONTInuous**

Description This command enables or disables the continuous rate of automated line error injection.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:CONTInuous <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:CONTInuous**

Parameter(s)

Set:

The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated line error injection continuously.

Example(s)

* SOUR:DATA:TEL:SON:ERR:LINE:AUT:TYPE
BERR
* SOUR:DATA:TEL:SON:ERR:LINE:AUT:CONT ON
* SOUR:DATA:TEL:SON:ERR:LINE:AUT:CONT?
Returns 1
* SOUR:DATA:TEL:SON:ERR:LINE:AUT ON

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:CONTInuous?**

Description	This query returns the status of continuous rate of automated Line error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:CONTInuous?**

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuous rate of automated line error injection.

Example(s)

* SOUR:DATA:TEL:SON:ERR:LINE:AUT:TYPE
BERR
* SOUR:DATA:TEL:SON:ERR:LINE:AUT:CONT ON
* SOUR:DATA:TEL:SON:ERR:LINE:AUT:CONT?
Returns 1
* SOUR:DATA:TEL:SON:ERR:LINE:AUT ON

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:AUTomated:CONTInuous

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:TYPE

Description	<p>This command selects the Burst type for the section error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:TYPE<wsp>BERRor FAS
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS. Selects the Burst type for the section error. BERRor, selects B1 (BERROR) as section error. FAS, selects Frame Alignment Signal (FAS) as section error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:TYPE? Returns BERROR</p>
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:BURSt:TYPE?**

Description	<p>This query returns the Burst type for the section error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the section error. BERROR, B1 (BERROR) is selected as section error. FAS, Frame Alignment Signal (FAS) is selected as section error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:TYPE? Returns BERROR</p>
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:TYPE

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:MODE

Description	<p>This command selects the Burst mode for the section error.</p> <p>At *RST, this value is set to SINGLE.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:MODE <wsp>SINGle REPeat
Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat.</p> <p>Selects the Burst mode for the section error.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:MODE? Returns SINGLE</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTIon:BURSt:MODE?**

Description This query returns the Burst mode for the section error.

At *RST, this value is set to SINGLE.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTIon:BURSt:MODE?

Parameter(s) None

Response Syntax <Mode>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:BURSt:MODE?**

Response(s)	Mode: The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the section error. SINGLE, Single is selected as Burst mode. REPEAT, Repeat is selected as Burst mode.
Example(s)	* SOUR:DATA:TEL:SON:ERR:SECT:BURS:MODE SING * SOUR:DATA:TEL:SON:ERR:SECT:BURS:MODE? Returns SINGLE
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:MODE

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt:DURation**

Description This command sets the duration for the section error.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt:DURation<wsp> <Duration>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
SECTION:BURSt:DURation**

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the section error. Choices are 1 through 14400000.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:DUR 15</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:DUR? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERROr: SECTION:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERROr: SECTION:BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERROr: SECTION:BURSt:DURation?</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:DURation?

Description	<p>This query returns the duration for the section error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:DURation? [<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<p><Duration></p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt:DURation?**

Response(s)	Duration: The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the duration for section error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:SECT:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:SECT:BURS:MODE SING * SOUR:DATA:TEL:SON:ERR:SECT:BURS:DUR 15 * SOUR:DATA:TEL:SON:ERR:SECT:BURS:DUR? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:DURation

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt:PERiod**

Description

This command sets the period for the section error.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt:PERiod <wsp> <Period>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
SECTION:BURSt:PERiod**

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for section error. Choices are 1 through 14400000.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:SECT:BURS:TYPE BERR* SOUR:DATA:TEL:SON:ERR:SECT:BURS:MODE REP* SOUR:DATA:TEL:SON:ERR:SECT:BURS:PER 15* SOUR:DATA:TEL:SON:ERR:SECT:BURS:PER? Returns 15
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERROr:SECTION:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERROr:SECTION:BURSt:MODE* SOURce[1..n]:DATA:TELEcom:SONet:ERROr:SECTION:BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt:PERiod?**

Description	<p>This query returns the period for the section error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:PERiod?[<wsp> MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<p><Period></p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTion:BURSt:PERiod?**

Response(s)	Period: The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the period for the section error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:SECT:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:SECT:BURS:MODE REP * SOUR:DATA:TEL:SON:ERR:SECT:BURS:PER 15 * SOUR:DATA:TEL:SON:ERR:SECT:BURS:PER? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTion:BURSt:PERiod

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt**

Description This command enables or disables the Burst for the section error.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt<wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the section error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS:DUR 10</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS ON</p> <p>* SOUR:DATA:TEL:SON:ERR:SECT:BURS</p> <p>Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:DURation</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt?**

Description	This query returns the status of Burst for the section error. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
SECTION:BURSt?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Burst for the section error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:SECT:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:SECT:BURS:DUR 10 * SOUR:DATA:TEL:SON:ERR:SECT:BURS ON * SOUR:DATA:TEL:SON:ERR:SECT:BURS Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt:DURation * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: SECTION:BURSt

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTion:BURSt:TYPE**

Description	<p>This command selects the Burst type for the section alarm.</p> <p>At *RST, this value is set to LOF1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion:BURSt:TYPE<wsp>LOF1 SEF1</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF1 SEF1.</p> <p>Selects the Burst type for the section alarm.</p> <p>LOF1, selects Loss of Frame (LOF1) as section alarm.</p> <p>SEF1,selects Severely Errored Framing (SEF1) as section alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE SEF1</p> <p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE? Returns SEF1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTion:BURSt:TYPE?**

Description	This query returns the Burst type for the section alarm. At *RST, this value is set to LOF1.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion:BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt:TYPE?

Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the section alarm. LOF1, Loss of Frame (LOF1) is selected as section alarm. SEF1, Errored Framing (SEF1) is selected as section alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE SEF1</p> <p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE? Returns SEF1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:MODE****Description**

This command selects the Burst mode for the section alarm.

At *RST, this value is set to SINGLE.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:MODE <wsp>SINGLE | REPEAT

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:MODE**

Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat. Selects the Burst mode for the section alarm.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE SEF1</p> <p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:MODE? Returns SINGLE</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTION:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTION:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTion:BURSt:MODE?**

Description	This query returns the Burst mode for the section alarm. At *RST, this value is set to SINGLE.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion:BURSt:MODE?
Parameter(s)	None
Response Syntax	<Mode>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt:MODE?

Response(s) Mode:
The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the section alarm. SINGLE, Single is selected as Burst mode. REPEAT, Repeat is selected as Burst mode.

Example(s) * SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE
SEF1
* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:MODE
SING
* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:
MODE? Returns SINGLE

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:MODE

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTIon:BURSt:DURation**

Description	<p>This command sets the duration for the section alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt:DURation<wsp><Duration> MAXimum MINimum</pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:DURation**

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the section alarm. Choices are 1 through 14400000.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE SEF1</p> <p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:DUR 15</p> <p>* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:DUR? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTION:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTION:BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTION:BURSt:DURation?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:DURation?**

Description	<p>This query returns the duration for the section alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt:DURation?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<pre><Duration></pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:DURation?**

Response(s)	Duration: The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the duration for the section alarm.
Example(s)	* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE SEF1 * SOUR:DATA:TEL:SON:ALAR:SECT:BURS:MODE SING * SOUR:DATA:TEL:SON:ALAR:SECT:BURS:DUR 15 * SOUR:DATA:TEL:SON:ALAR:SECT:BURS:DUR? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt:DURation

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:PERiod****Description**

This command sets the period for the section alarm.

At *RST, this value is set to 1.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:  
SECTION:BURSt:PERiod <wsp> <Period>  
|MAXimum|MINimum
```

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:PERiod**

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the section alarm. Choices are 1 through 14400000.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE SEF1* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:MODE REP* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:PER 15* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:PER? Returns 15
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTION:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTION:BURSt:MODE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTION:BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:PERiod?**

Description	<p>This query returns the period for the section alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt:PERiod?[<wsp> MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<pre><Period></pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTion:BURSt:PERiod?**

Response (s)	Period: The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the period for the section alarm.
Example(s)	* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE SEF1 * SOUR:DATA:TEL:SON:ALAR:SECT:BURS:MODE REP * SOUR:DATA:TEL:SON:ALAR:SECT:BURS:PER 15 * SOUR:DATA:TEL:SON:ALAR:SECT:BURS:PER? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTion:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTion:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm:SECTion:BURSt:PERiod

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTIon:BURSt**

Description	<p>This command enables or disables the Burst for the section alarm.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTIon:BURSt<wsp> <Set></pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt**

Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the section alarm.
Example(s)	* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE SEF1 * SOUR:DATA:TEL:SON:ALAR:SECT:BURS: DUR 10 * SOUR:DATA:TEL:SON:ALAR:SECT:BURS ON * SOUR:DATA:TEL:SON:ALAR:SECT:BURS Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt:DURation * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTion:BURSt?**

Description	This query returns the status of the Burst for the section alarm. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTion:BURSt?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: SECTION:BURSt?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Burst for the section alarm.

Example(s)

* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:TYPE
SEF1

* SOUR:DATA:TEL:SON:ALAR:SECT:BURS:
DUR 10

* SOUR:DATA:TEL:SON:ALAR:SECT:BURS ON

* SOUR:DATA:TEL:SON:ALAR:SECT:BURS

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt:DURation

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
SECTION:BURSt

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:TYPE**

Description	<p>This command selects the Burst type for the line error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE<wsp>BERRor REI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the Burst type for the line error.</p> <p>BERRor, selects B2 (BERRor) as line error.</p> <p>REI, selects Remote Error Indication - Line (REI) as line error.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE? Returns BERROR</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE?</pre>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE?

Description This query returns the Burst type for the line error.

At *RST, this value is set to BERRor.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:TYPE?

Parameter(s) None

Response Syntax <Error>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:TYPE?**

Response(s)	Error: The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the line error. BERROR, B2 (BERRor) is selected as line error. REI, Remote Error Indication - Line (REI) is selected as line error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE? Returns BERROR
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:MODE

Description This command selects the Burst mode for the line error.

At *RST, this value is set to SINGLE.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:MODE<wsp>SINGLE|REPeat

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:MODE

Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat.</p> <p>Selects the Burst mode for the line error.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:LINE:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SON:ERR:LINE:BURS:MODE? Returns SINGLE</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:MODE?**

Description This query returns the Burst mode for the line error.

At *RST, this value is set to SINGLE.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:MODE?

Parameter(s) None

Response Syntax <Mode>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:MODE?**

Response(s)	Mode: The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the line error. SINGLE, Single is selected as Burst mode. REPEAT, Repeat is selected as Burst mode.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:BURS:MODE SING * SOUR:DATA:TEL:SON:ERR:LINE:BURS:MODE? Returns SINGLE
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:MODE

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:DURation**

Description This command sets the duration for the line error.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:DURation <wsp> <Duration> |
MAXimum | MINimum

:SOURce[1..n]:DATA:TELEcom:SONet:ERror: LINE:BURSt:DURation

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the line error. Choices are 1 through 14400000.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:BURS:MODE SING * SOUR:DATA:TEL:SON:ERR:LINE:BURS:DUR 15 * SOUR:DATA:TEL:SON:ERR:LINE:BURS:DUR? Returns 15</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SONet:ERror: LINE:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERror: LINE:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ERror: LINE:BURSt:DURation?</pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:DURation?**

Description	<p>This query returns the duration for the line error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:DURation?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<pre><Duration></pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:DURation?**

Response(s)	Duration: The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the duration for the line error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:BURS:MODE SING * SOUR:DATA:TEL:SON:ERR:LINE:BURS:DUR 15 * SOUR:DATA:TEL:SON:ERR:LINE:BURS:DUR? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:DURation

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:PERiod**

Description This command sets the period for the line error.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:PERiod <wsp> <Period>
|MAXimum|MINimum

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:PERiod

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the line error. Choices are 1 through 14400000.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:BURS:MODE REP * SOUR:DATA:TEL:SON:ERR:LINE:BURS:PER 15 * SOUR:DATA:TEL:SON:ERR:LINE:BURS:PER? Returns 15</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:PERiod?</pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:PERiod?**

Description	<p>This query returns the period for the line error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:PERiod?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<p><Period></p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt:PERiod?**

Response(s)	Period: The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the period for the line error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:BURS:MODE REP * SOUR:DATA:TEL:SON:ERR:LINE:BURS:PER 15 * SOUR:DATA:TEL:SON:ERR:LINE:BURS:PER? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:PERiod

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt

Description This command enables or disables the Burst for the line error.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt**

Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the line error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:BURS:DUR 10 * SOUR:DATA:TEL:SON:ERR:LINE:BURS ON * SOUR:DATA:TEL:SON:ERR:LINE:BURS Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:DURation * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt?

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt?

Description	This query returns the status of Burst for the line error. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LINE:BURSt?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Burst for the line error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:LINE:BURS:DUR 10 * SOUR:DATA:TEL:SON:ERR:LINE:BURS ON * SOUR:DATA:TEL:SON:ERR:LINE:BURS Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt:DURation * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LINE:BURSt

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:TYPE**

Description	<p>This command selects the Burst type for the line alarm.</p> <p>At *RST, this value is set to AIS.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE<wsp>AIS RDI</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS RDI.</p> <p>Selects the Burst type for the line alarm.</p> <p>AIS, selects Alarm Indication Signal - Line (AIS) as line alarm.</p> <p>RDI, selects Remote Defect Indication - Line (RDI) as line alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:TYPE? Returns AIS</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:TYPE?**

Description	This query returns the Burst type for the line alarm. At *RST, this value is set to AIS.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE?

Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of line alarm.</p> <p>AIS, Alarm Indication Signal - Line (AIS) is selected as line alarm.</p> <p>RDI, Remote Defect Indication - Line (RDI) is selected as line alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:TYPE? Returns AIS</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:MODE****Description**

This command selects the Burst mode for the line alarm.

At *RST, this value is set to SINGLE.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:MODE<wsp>SINGLE|REPeat

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:MODE**

Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat.</p> <p>Selects the Burst mode for the line alarm.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:MODE? Returns SINGLE</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:MODE?**

Description	This query returns the Burst mode for the line alarm. At *RST, this value is set to SINGLE.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:MODE?
Parameter(s)	None
Response Syntax	<Mode>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:MODE?

Response(s)	<p>Mode:</p> <p>The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the line alarm. SINGLE, Single is selected as Burst mode. REPEAT, Repeat is selected as Burst mode.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:MODE SING* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:MODE? Returns SINGLE
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:MODE

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:DURation**

Description	<p>This command sets the duration for the line alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:DURation <wsp> <Duration> MAXimum MINimum</pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:DURation**

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the line alarm. Choices are 1 through 14400000.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ERR:LINE:BURS:DUR 15</p> <p>* SOUR:DATA:TEL:SON:ERR:LINE:BURS:DUR?</p> <p>Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LINE:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LINE:BURSt:DURation?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:DURation?**

Description	<p>This query returns the duration for the line alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:DURation?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<pre><Duration></pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:DURation?**

Response(s)	Duration: The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the duration for the line alarm.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LINE:BURS:TYPE AIS * SOUR:DATA:TEL:SON:ERR:LINE:BURS:DUR 15 * SOUR:DATA:TEL:SON:ERR:LINE:BURS:DUR? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:DURation

**:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt**

Description	<p>This command enables or disables the Burst for the line alarm.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the line alarm.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:TYPE AIS * SOUR:DATA:TEL:SON:ALAR:LINE:BURS ON * SOUR:DATA:TEL:SON:ALAR:LINE:BURS? Returns 1</pre>
See Also	<pre>* SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE * SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt?</pre>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt?

Description	This query returns the status of Burst for the line alarm. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt?
Parameter(s)	None
Response Syntax	<Set>

**:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Burst for the line alarm.
Example(s)	* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:TYPE AIS * SOUR:DATA:TEL:SON:ALAR:LINE:BURS ON * SOUR:DATA:TEL:SON:ALAR:LINE:BURS? Returns 1
See Also	* SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE * SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:PERiod**

Description This command sets the period for the line alarm.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:PERiod <wsp> <Period>
|MAXimum|MINimum

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:PERiod

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the line alarm. Choices are 1 through 14400000.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ALAR:LINE:BURS:MODE REP * SOUR:DATA:TEL:SON:ALAR:LINE:BURS:PER 15 * SOUR:DATA:TEL:SON:ALAR:LINE:BURS:PER? Returns 15</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:PERiod?</pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:PERiod?**

Description This query returns the period for the line alarm.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:PERiod?

Parameter(s) The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum|MINimum.
MAXimum is used to retrieve the instrument's greatest supported value.
MINimum is used to retrieve the instrument's smallest supported value.
This parameter is optional. If no token is specified, the current period will be returned.

Response Syntax <Period>

**:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm:
LINE:BURSt:PERiod?**

Response(s)	Period: The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the period for the line alarm.
Example(s)	* SOUR:DATA:TEL:SON:ALAR:LINE:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ALAR:LINE:BURS:MODE REP * SOUR:DATA:TEL:SON:ALAR:LINE:BURS:PER 15 * SOUR:DATA:TEL:SON:ALAR:LINE:BURS:PER? Returns 15
See Also	* SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:TYPE * SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:MODE * SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LINE:BURSt:PERiod

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
HOP:PATH:BURSt:TYPE**

Description	<p>This command selects the Burst type for the High Order Path (HOP) error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: HOP:PATH:BURSt:TYPE<wsp>BERRor REI</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the Burst type for the High Order Path (HOP) error.</p> <p>BERRor, selects B3 (BERROR) as HOP error.</p> <p>REI, selects Remote Error Indication - Path (REI) as HOP error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS: TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS: TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERROr: HOP:PATH:BURSt:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:TYPE?**

Description	This query returns the Burst type for the High Order Path (HOP) error. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the High Order Path (HOP) error.</p> <p>BERROR, B3 (BERROR) is selected as HOP error. REI, Remote Error Indication - Path (REI) is selected as HOP error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:BURSt:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:MODE****Description**

This command selects the Burst mode for the High Order Path (HOP) error.

At *RST, this value is set to SINGLE.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:MODE<wsp>SINGle|REPeat

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:MODE**

Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPEAT.</p> <p>Selects the Burst mode for the High Order Path (HOP) error.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPEAT, selects Repeat as Burst mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:MODE? Returns SINGLE</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:MODE?**

Description	This query returns the Burst mode for the High Order Path (HOP) error. At *RST, this value is set to SINGLE.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:MODE?
Parameter(s)	None
Response Syntax	<Mode>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:MODE?

Response(s)	<p>Mode:</p> <p>The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:TYPE BERR* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:MODE SING* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:MODE? Returns SINGLE
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:BURSt:MODE

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:DURation****Description**

This command sets the duration for the High Order Path (HOP) error.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:DURation <wsp> <Duration>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:DURation**

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the High Order Path (HOP) error. Choices are 1 through 14400000.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:DUR 15</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:DUR? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:BURSt:DURation?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:DURation?**

Description	<p>This query returns the duration for the High Order Path (HOP) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:DURation?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<pre><Duration></pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:DURation?**

Response(s)	Duration: The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the duration for the High Order Path (HOP) error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS: TYPE BERR * SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS: DUR 15 * SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS: DUR? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:DURation

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:PERiod**

Description This command sets the period for the High Order Path (HOP) error.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:PERiod <wsp> <Period>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:PERiod**

Parameter(s) Period:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the High Order Path (HOP) error. Choices are 1 through 14400000.

Example(s) * SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:
TYPE BERR
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:
MODE REP
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:
PER 15
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:
PER? Returns 15

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:MODE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:PERiod?**

Description	<p>This query returns the period for the High Order Path (HOP) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:PERiod?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<pre><Period></pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt:PERiod?**

Response(s)	Period: The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the period for the High Order Path (HOP) error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:TYPE BERR * SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:MODE REP * SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:PER 15 * SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS:PER? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:BURSt:PERiod

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt****Description**

This command enables or disables the Burst for the High Order Path (HOP) error.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt<wsp><Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt**

Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the High Order Path (HOP) error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURSt: TYPE BERR * SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURSt ON * SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURSt? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:BURSt?**

Description	This query returns the status of Burst for the High Order Path (HOP) error. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt?

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Burst for the High Order Path (HOP) error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURSt: TYPE BERR * SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS ON * SOUR:DATA:TEL:SON:ERR:HOP:PATH:BURS? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:BURSt

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:TYPE**

Description	<p>This command selects the Burst type for the High Order Path (HOP) alarm.</p> <p>At *RST, this value is set to AIS.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:TYPE<wsp>AIS RDI EPSD1 EPCD1 EPPD1 LOM LOP PDI UNEQP1</pre>

:SOURCE[1..n]:DATA:TELEcom:SONet:ALARM: HOP:PATH:BURSt:TYPE

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

AIS|RDI|EPSD1|EPCD1|EPPD1|LOM|LOP|PDI|UNEQP1.

Selects the Burst type for the High Order Path (HOP) alarm.

AIS, selects Alarm Indication Signal (AIS) as HOP alarm.

RDI, selects Remote Defect Indication test (RDI) as HOP alarm.

EPSD1, selects Enhanced RDI - Path Server Defect (EPSD1) as HOP alarm.

EPCD1, selects Enhanced RDI - Path Connectivity Defect (EPCD1) as HOP alarm.

EPPD1, selects Enhanced RDI - Path Payload Defect (EPPD1) as HOP alarm.

LOM, selects Loss of Multiframe (LOM) as HOP alarm.

LOP, selects Low Order Path (LOP) as HOP alarm.

PDI, selects Payload Defect Indication - Path (PDI) as HOP alarm.

UNEQP1, selects Unequipped - Path (UNEQP1) as HOP alarm.

**:SOURce[1..n]:DATA:TELecom:SONet:ALARm:
HOP:PATH:BURSt:TYPE**

Example(s)	* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS: TYPE AIS * SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS: TYPE? Returns AIS
See Also	* SOURce[1..n]:DATA:TELecom:SONet:ALARm: HOP:PATH:BURSt:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:TYPE?**

Description	This query returns the Burst type for the High Order Path (HOP) alarm. At *RST, this value is set to AIS.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:TYPE?**

Response(s)

Alarm:

The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Burst type for the High Order Path (HOP) alarm.

AIS, Alarm Indication Signal (AIS) is selected as HOP alarm.

RDI, Remote Defect Indication test (RDI) is selected as HOP alarm.

EPsD1, Enhanced RDI - Path Server Defect (EPsD1) is selected as HOP alarm.

EPcD1, Enhanced RDI - Path Connectivity Defect (EPcD1) is selected as HOP alarm.

EPpD1, Enhanced RDI - Path Payload Defect (EPpD1) is selected as HOP alarm.

LOM, Loss of Multiframe (LOM) is selected as HOP alarm.

LOP, Low Order Path (LOP) is selected as HOP alarm.

PDI, Payload Defect Indication - Path (PDI) is selected as HOP alarm.

UNEQP1, Unequipped - Path (UNEQP1) is selected as HOP alarm.

Example(s)

* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:TYPE AIS

* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:TYPE? Returns AIS

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:TYPE

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:MODE**

Description

This command selects the Burst mode for the High Order Path (HOP) alarm.

At *RST, this value is set to SINGLE.

Syntax

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:MODE<wsp>SINGle | REPeat

:SOURce[1..n]:DATA:TELecom:SONet:ALARm: HOP:PATH:BURSt:MODE

Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat.</p> <p>Selects the Burst mode for the High Order Path (HOP) alarm.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:MODE? Returns SINGLE</p>
See Also	<p>* SOURce[1..n]:DATA:TELecom:SONet:ALARm:HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELecom:SONet:ALARm:HOP:PATH:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:MODE?**

Description This query returns the Burst mode for the High Order Path (HOP) alarm.

At *RST, this value is set to SINGLE.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:MODE?

Parameter(s) None

Response Syntax <Mode>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:MODE?**

Response(s)	<p>Mode:</p> <p>The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. 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>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:MODE? Returns SINGLE</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:MODE</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:DURation

Description	<p>This command sets the duration for the High Order Path (HOP) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:DURation <wsp> <Duration> MAXimum MINimum</pre>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:DURation

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the High Order Path (HOP) alarm. Choices are 1 through 14400000.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS: TYPE AIS * SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS: MODE SING * SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS: DUR 15 * SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS: DUR? Returns 15</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:DURation?</pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:DURation?**

Description	<p>This query returns the duration for the High Order Path (HOP) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:DURation?[<wsp> MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<p><Duration></p>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:DURation?

Response(s)	<p>Duration:</p> <p>The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the duration for the High Order Path (HOP) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:DUR 15</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:DUR? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:DURation</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:PERiod**

Description This command sets the period for the High Order Path (HOP) alarm.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:PERiod <wsp> <Period>
|MAXimum|MINimum

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:PERiod

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the High Order Path (HOP) alarm. Choices are 1 through 14400000.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:MODE REP</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:PER 15</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:PER? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:PERiod?</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt:PERiod?**

Description	<p>This query returns the period for the High Order Path (HOP) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:PERiod?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<p><Period></p>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:PERiod?

Response(s)	<p>Period:</p> <p>The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the period for the High Order Path (HOP) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:TYPE AIS</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:MODE REP</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:PER 15</p> <p>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:PER? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:PERiod</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt

Description This command enables or disables the Burst for the High Order Path (HOP) alarm.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt<wsp><Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the High Order Path (HOP) alarm.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS ON* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:HOP:PATH:BURSt?

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt?

Description This query returns the status of Burst for the High Order Path (HOP) alarm.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt?

Parameter(s) None

Response Syntax <Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:BURSt?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Burst for the High Order Path (HOP) alarm.
Example(s)	* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS: TYPE AIS * SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS ON * SOUR:DATA:TEL:SON:ALAR:HOP:PATH:BURS Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:BURSt

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:TYPE**

Description	<p>This command selects the Burst type for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to BIP2.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:TYPE<wsp>BIP2 REI</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 REI.</p> <p>Selects the Burst type for the Low Order Path (LOP) error.</p> <p>BIP2, selects Bit-Interleaved Parity - 2 bits (BIP2) as LOP error.</p> <p>REI, selects Remote Error Indication (REI) as LOP error.</p>

**:SOURce[1..n]:DATA:TELecom:SONet:ERRor:
LOP:PATH:BURSt:TYPE**

Example(s)	* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS: TYPE BIP2 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS: TYPE? Returns BIP2
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELecom:SONet:ERRor: LOP:PATH:BURSt:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROR:
LOP:PATH:BURSt:TYPE?**

Description	<p>This query returns the Burst type for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to BIP2.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERROR: LOP:PATH:BURSt:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the Low Order Path (LOP) error.</p> <p>BIP2, Bit-Interleaved Parity - 2 bits (BIP2) is selected as LOP error.</p> <p>REI, Remote Error Indication (REI) is selected as LOP error.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:TYPE?**

Example(s)	* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS: TYPE BIP2 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS: TYPE? Returns BIP2
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:TYPE

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:MODE**

Description	<p>This command selects the Burst mode for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to SINGLE.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:MODE<wsp>SINGLE REPeat</p>
Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGLE REPeat.</p> <p>Selects the Burst mode for the Low Order Path (LOP) error.</p> <p>SINGLE, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:MODE****Example(s)**

* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:
TYPE BIP2
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:
MODE SING
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:
MODE? Returns SINGLE

Note

FTB/IQS-8140 Transport Blazer does not support
this command.

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:MODE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LOP:PATH:BURSt:MODE?**

Description	<p>This query returns the Burst mode for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to SINGLE.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LOP:PATH:BURSt:MODE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Mode></p>
Response(s)	<p>Mode: The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the Low Order Path (LOP) error. SINGLE, selects Single as Burst mode. REPEAT, selects Repeat as Burst mode.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:MODE?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:TYPE BIP2* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:MODE SING* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:MODE? Returns SINGLE
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:MODE

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LOP:PATH:BURSt:DURation**

Description	<p>This command sets the duration for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LOP:PATH:BURSt:DURation<wsp><Duration> MAXimum MINimum</p>
Parameter(s)	<p>Duration: The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the Low Order Path (LOP) error. Choices are 1 through 14400000.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:DURation****Example(s)**

* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:
TYPE BIP2
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:
MODE SING
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:
DUR 15
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:
DUR? Returns 15

Note

FTB/IQS-8140 Transport Blazer does not support
this command.

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:MODE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:DURation?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:DURation?**

Description	<p>This query returns the duration for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:DURation?[<wsp> MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<p><Duration></p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:DURation?**

Response(s)	Duration: The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the duration for the Low Order Path (LOP) error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:TYPE BIP2 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:MODE SING * SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:DUR 15 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:DUR? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:DURation

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:PERiod**

Description This command sets the period for the Low Order Path (LOP) error.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:PERiod <wsp> <Period>
|MAXimum|MINimum

Parameter(s) Period:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum|MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the Low Order Path (LOP) error. Choices are 1 through 14400000.

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt:PERiod**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:TYPE BIP2* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:MODE REP* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:PER 15* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:PER? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:MODE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LOP:PATH:BURSt:PERiod?**

Description	<p>This query returns the period for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LOP:PATH:BURSt:PERiod? [<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<p><Period></p>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt:PERiod?

Response(s)	<p>Period:</p> <p>The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the period for the Low Order Path (LOP) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:TYPE BIP2</p> <p>* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:MODE REP</p> <p>* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:DUR 15</p> <p>* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:DUR? Returns 15</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:PERiod</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt

Description	<p>This command enables or disables Burst for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt <wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables Burst for the Low Order Path (LOP) error.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:TYPE BIP2* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURSON* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt?**

Description	<p>This query returns the status of Burst for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:BURSt?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Burst for the Low Order Path (LOP) error.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:BURSt?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS:TYPE BIP2* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURSON* SOUR:DATA:TEL:SON:ERR:LOP:PATH:BURS?Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:BURSt

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:TYPE**

Description This command selects the Burst type for the Low Order Path (LOP) alarm.

At *RST, this value is set to AIS.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:TYPE<wsp>AIS|RDI|EVSD|
EVCD|EVPD|RFIV|LOP|UNEQv

**:SOURce[1..n]:DATA:TELecom:SONet:ALARm:
LOP:PATH:BURSt:TYPE**

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

AIS|RDI|EVSD|EVCD|EVPD|RFIV|LOP|UNEQv.

Selects the Burst type for the Low Order Path (LOP) alarm.

AIS, selects Alarm Indication Signal (AIS) as LOP alarm.

RDI, selects Remote Defect Indication test (RDI) as LOP alarm.

EVSD, selects Enhanced RDI - VT Server Defect (EVSD) as LOP alarm.

EVCD, selects Enhanced RDI - VT Connectivity Defect (EVCD) as LOP alarm.

EVPD, selects Enhanced RDI - VT Payload Defect (EVPD) as LOP alarm.

RFIV, selects Remote Failure Indication - VT (RFIV) as LOP alarm.

LOP, selects Low Order Path (LOP) as LOP alarm.

UNEQv, selects Unequipped - VT (UNEQv) as LOP alarm.

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:TYPE

Example(s)	* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS: TYPE AIS * SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS: TYPE? Returns AIS
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:TYPE?**

Description	This query returns the Burst type for the Low Order Path (LOP) alarm. At *RST, this value is set to AIS.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>

:SOURce[1..n]:DATA:TELecom:SONet:ALARm: LOP:PATH:BURSt:TYPE?

Response(s)

Alarm:

The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Burst type for the Low Order Path (LOP) alarm.

AIS, Alarm Indication Signal (AIS) is selected as LOP alarm.

RDI, Remote Defect Indication test (RDI) is selected as LOP alarm.

EVSD, Enhanced RDI - VT Server Defect (EVSD) is selected as LOP alarm.

EVCD, Enhanced RDI - VT Connectivity Defect (EVCD) is selected as LOP alarm.

EVPD, Enhanced RDI - VT Payload Defect (EVPD) is selected as LOP alarm.

RFIV, Remote Failure Indication - VT (RFIV) is selected as LOP alarm.

LOP, Low Order Path (LOP) is selected as LOP alarm.

UNEQV, Unequipped - VT (UNEQV) is selected as LOP alarm.

**:SOURce[1..n]:DATA:TELecom:SONet:ALARm:
LOP:PATH:BURSt:TYPE?**

Example(s)	* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS: TYPE AIS * SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS: TYPE? Returns AIS
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELecom:SONet:ALARm: LOP:PATH:BURSt:TYPE

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:MODE**

Description	<p>This command selects the Burst mode for the Low Order Path (LOP) alarm.</p> <p>At *RST, this value is set to SINGLE.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:MODE<wsp>SINGLE REPeat</p>
Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGLE REPeat.</p> <p>Selects the Burst mode for the Low Order Path (LOP) alarm.</p> <p>SINGLE, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:MODE**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:MODE SING* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:MODE? Returns SINGLE
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:MODE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:MODE?**

Description	<p>This query returns the Burst mode for the Low Order Path (LOP) alarm.</p> <p>At *RST, this value is set to SINGLE.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:MODE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Mode></p>
Response(s)	<p>Mode: The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the Low Order Path (LOP) alarm. SINGLE, Single is selected as Burst mode. REPEAT, Repeat is selected as Burst mode.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:MODE?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:MODE SING* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:MODE? Returns SINGLE
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:MODE

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:DURation**

Description	<p>This command sets the duration for the Low Order Path (LOP) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:DURation<wsp><Duration> MAXimum MINimum</pre>
Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the Low Order Path (LOP) alarm. Choices are 1 through 14400000.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:DURation**

Example(s) * SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:
TYPE AIS
* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:
MODE SING
* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:
DUR 15
* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:
DUR? Returns 15

Note FTB/IQS-8140 Transport Blazer does not support
this command.

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:MODE
* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:DURation?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:DURation?**

Description	<p>This query returns the duration for the Low Order Path (LOP) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt:DURation?[<wsp> MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<p><Duration></p>

**:SOURce[1..n]:DATA:TELecom:SONet:ALARm:
LOP:PATH:BURSt:DURation?**

Response(s)	<p>Duration:</p> <p>The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the duration for the Low Order Path (LOP) alarm.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:MODE SING* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:DUR 15* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:DUR? Returns 15
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELecom:SONet:ALARm:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELecom:SONet:ALARm:LOP:PATH:BURSt:MODE* SOURce[1..n]:DATA:TELecom:SONet:ALARm:LOP:PATH:BURSt:DURation

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:PERiod**

Description This command sets the period for the Low Order Path (LOP) alarm.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:PERiod <wsp> <Period>
|MAXimum|MINimum

Parameter(s) Period:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum|MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the Low Order Path (LOP) alarm. Choices are 1 through 14400000.

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:PERiod**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:MODE REP* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:PER 15* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:PER? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:MODE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:PERiod?**

Description This query returns the period for the Low Order Path (LOP) alarm.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:PERiod? [<wsp>MAXimum |
MINimum]

Parameter(s) The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum | MINimum.
MAXimum is used to retrieve the instrument's greatest supported value.
MINimum is used to retrieve the instrument's smallest supported value.
This parameter is optional. If no token is specified, the current period will be returned.

Response Syntax <Period>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt:PERiod?**

Response(s)	<p>Period:</p> <p>The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the period for the Low Order Path (LOP) alarm.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:MODE REP* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:PER 15* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:PER? Returns 15
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:MODE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:PERiod

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt**

Description This command enables or disables the Burst for the Low Order Path (LOP) alarm.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt <wsp> <Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the Low Order Path (LOP) alarm.

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:MODE AIS* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURSON* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt?**

Description	This query returns the status of Burst for the Low Order Path (LOP) alarm. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:BURSt?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Burst for the Low Order Path (LOP) alarm.

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:BURSt?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURS:MODE AIS* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURSON* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:BURSReturns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:LOP:PATH:BURSt

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:TYPE

Description	<p>This command selects the type of High Order Path (HOP) alarm.</p> <p>At *RST, this value is set to AIS.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:TYPE<wsp>AIS RDI EPSD1 EPCD1 EPPD1 LOM LOP PDI UNEQP1</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS RDI EPSD1 EPCD1 EPPD1 LOM LOP PDI UNEQP1.</p> <p>Selects the type of HOP (High Order Path) alarm.</p> <p>AIS, selects the AIS-P (Alarm Indication Signal - Path) which generates all-ones pattern for the path and payload.</p> <p>RDI, selects the RDI-P (Remote Defect Indication - Path) which generates a "100" pattern for bits 5, 6 and 7 of the G1 byte.</p>

**:SOURCE[1..n]:DATA:TELEcom:SONet:ALARM:
HOP:PATH:TYPE**

EPD1, selects the ERDI-PSD (Enhanced RDI - Path Server Defect) which generates a "101" pattern for bits 5, 6 and 7 of the G1 byte.

EPCD1, selects the ERDI-PCD (Enhanced RDI - Path Connectivity Defect) which generates a "110" pattern for bits 5, 6 and 7 of the G1 byte.

EPPD1, selects the ERDI-PPD (Enhanced RDI - Path Payload Defect) which generates a "B010" pattern for bits 5, 6 and 7 of the G1 byte.

LOM, selects the Loss of Multiframe which generates a wrong H4 byte multiframe indicator sequence.

LOP, selects the LOP-P (Loss of Pointer - Path) which generates a non-valid pointer.

PDI, selects the PDI-P (Payload Defect Indication - Path) which generates a VT-structured STS-1 SPE with payload defect.

UNEQ1, selects the UNEQ-P (Unequipped - Path) which generates samples of unequipped STS signal labels (path and payload are set to "00 H").

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:TYPE

Example(s) * SOUR:DATA:TEL:SON:ALAR:HOP:PATH:TYPE
 AIS
 * SOUR:DATA:TEL:SON:ALAR:HOP:PATH:TYPE?
 Returns AIS

Note For **8120NGE/8130NGE/8130NGEv2** modules,
 choices are AIS|RDI|EPSD1|EPCD1|EPPD1|
 LOM|LOP|PDI|UNEQP1.
 For **8140 module**, choices are AIS|RDI|EPSD1|
 EPCD1|EPPD1|LOP|PDI|UNEQP1.

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
 HOP:PATH:TYPE?

**:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:TYPE?**

Description	<p>This query returns the type of High Order Path (HOP) alarm.</p> <p>At *RST, this value is set to AIS.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of High Order Path (HOP) alarm. AIS, Alarm Indication Signal - Path (AIS-P) is selected as High Order Path (HOP) alarm. RDI, Remote Defect Indication - Path (RDI-P) is selected as High Order Path (HOP) alarm. EPSD1, Enhanced Remote Defect Indication - Path Server Defect (ERDI-PSD) is selected as High Order Path (HOP) alarm.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:TYPE?

EPCD1, Enhanced Remote Defect Indication - Path Connectivity Defect (ERDI-PCD) is selected as High Order Path (HOP) alarm.

EPPD1, Enhanced Remote Defect Indication - Path Payload Defect (ERDI-PPD) is selected as High Order Path (HOP) alarm.

LOM, Loss of Multiframe (LOM) is selected as High Order Path (HOP) alarm.

LOP, Loss Of Pointer - Path (LOP-P) is selected as High Order Path (HOP) alarm.

PDI, Payload Defect Indication - Path (PDIP) is selected as High Order Path (HOP) alarm.

UNEQP1, Unequipped - Path (UNEQP) is selected High Order Path (HOP) alarm.

Example(s)

* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:TYPE
AIS

* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:TYPE?
Returns AIS

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:TYPE

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH**

Description	<p>This command enables or disables the High Order Path (HOP) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the HOP (High Order Path) alarm generation.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:TYPE AIS * SOUR:DATA:TEL:SON:ALAR:HOP:PATH ON * SOUR:DATA:TEL:SON:ALAR:HOP:PATH? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH?</pre>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH?**

Description	This query returns the status of High Order Path (HOP) alarm generation. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of High Order Path (HOP) alarm generation.
Example(s)	* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:TYPE AIS * SOUR:DATA:TEL:SON:ALAR:HOP:PATH ON * SOUR:DATA:TEL:SON:ALAR:HOP:PATH? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:MANual:TYPE**

Description	<p>This command sets the manual type of High Order Path (HOP) error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:MANual:TYPE <wsp>BERRor REI</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the type of High Order Path (HOP) error. BERRor, selects the type of High Order Path (HOP) error as B3.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:MANual:TYPE**

REI, selects the REI-P (Remote Error Indicator - Path) error when bits 1 through 4 of the G1 byte contain one pattern from the following binary range: "#B0001" through "#B1000" (1 to 8).

Example(s)

* SOUR:DATA:TEL:SON:ERR:HOP:PATH:MAN:
TYPE BERR

* SOUR:DATA:TEL:SON:ERR:HOP:PATH:MAN:
TYPE? Returns BERROR

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:MANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:MANual:TYPE?**

Description	This query returns the manual type of High Order Path (HOP) error. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:MANual:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of High Order Path (HOP) error. BERROR, 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>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:MAN:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:MAN:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:MANual:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
HOP:PATH:AMOUnt**

Description This command sets the amount of High Order Path (HOP) error to be injected into the instrument.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
HOP:PATH:AMOUnt <wsp> <Amount>
|MAXimum|MINimum

Parameter(s) Amount:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AMOUNT**

Sets the amount of HOP (High Order Path) error to be injected.

Choices are 1 through 50.

Example(s)

* SOUR:DATA:TEL:SON:ERR:HOP:PATH:MAN:TYPE BERR

* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AMO 15

* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AMO?

Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:MANual:TYPE

* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:AMOUNT?

:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: HOP:PATH:AMOUnt?

Description This query returns the amount of High Order Path (HOP) error injected into the instrument.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
HOP:PATH:AMOUnt?[<wsp>MAXimum |
MINimum]

Parameter(s) The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum | MINimum.
MAXimum is used to retrieve the instrument's greatest supported value.
MINimum is used to retrieve the instrument's smallest supported value.
This parameter is optional. If no token is specified, the current amount of error will be returned.

Response Syntax <Amount>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AMOut?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of High Order Path (HOP) error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:HOP:PATH:MAN:TYPE BERR * SOUR:DATA:TEL:SON:ERR:HOP:PATH:AMO 15 * SOUR:DATA:TEL:SON:ERR:HOP:PATH:AMO? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:AMOut

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:INJect

Description	<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>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:INJect</pre>
Parameter(s)	None
Example(s)	<pre>* SOUR:DATA:TEL:SON:ERR:HOP:PATH: MAN: TYPE BERR * SOUR:DATA:TEL:SON:ERR:HOP:PATH: AMO 15 * SOUR:DATA:TEL:SON:ERR:HOP:PATH:INJ</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AMOUNT</pre>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERROr:
HOP:PATH:HISTory?**

Description	<p>This query returns the history status of High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ERROr:HOP: PATH:HISTory?<wsp>BERRor REI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the type of HOP (High Order Path) error. BERRor, selects the type of HOP (High Order Path) error as B3.</p> <p>REI, selects the REI-P (Remote Error Indicator - Path) error when bits 1 through 4 of the G1 byte contain one pattern from the following binary range: "#B0001" through "#B1000" (1 to 8).</p>
Response Syntax	<pre><History></pre>

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:HISTory?

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:HOP:PATH:MAN:TYPE BERR* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AMO 15* SOUR:DATA:TEL:SON:ERR:HOP:PATH:INJ* FETC:DATA:TEL:SON:ERR:HOP:PATH:HIST? BERR <p>Returns the error history.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:AMOUNT* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERROr:HOP:
PATH:SECOnds?**

Description This query returns the number of seconds within which High Order Path (HOP) error occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SONet:ERROr:HOP:
PATH:SECOnds? <wsp> BERRor|REI

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor|REI.
Selects the type of HOP (High Order Path) error.
BERRor, selects the type of HOP (High Order Path) error as B3.
REI, selects the REI-P (Remote Error Indicator - Path) error when bits 1 through 4 of the G1 byte contain one pattern from the following binary range: "#B0001" through "#B1000" (1 to 8).

Response Syntax <Seconds>

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:SEConds?

Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of seconds of High Order Path (HOP) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:MAN:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AMO 15</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:INJ</p> <p>* FETC:DATA:TEL:SON:ERR:HOP:PATH:SEC? BERR</p> <p>Returns the number of errored seconds.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:AMOUNT</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:INJect</p>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERROr:
HOP:PATH:CURRent?**

Description	<p>This query returns the current status of High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SONet:ERROr:HOP: PATH:CURRent?<wsp>BERRor REI</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the type of HOP (High Order Path) error. BERRor, selects the type of HOP (High Order Path) error as B3.</p> <p>REI, selects the REI-P (Remote Error Indicator - Path) error when bits 1 through 4 of the G1 byte contain one pattern from the following binary range: "#B0001" through "#B1000" (1 to 8).</p>
Response Syntax	<code><Current></code>

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:CURRent?

Response(s) Current:
The response data syntax for <Current> 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:SON:ERR:HOP:PATH:MAN:
TYPE BERR
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AMO 15
* SOUR:DATA:TEL:SON:ERR:HOP:PATH:INJ
* FETC:DATA:TEL:SON:ERR:HOP:PATH:CURR?
BERR
Returns the current error status.

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:AMOUNT
* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
HOP:PATH:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERROr:
HOP:PATH:COUNT?**

Description	<p>This query returns the count of High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SONet:ERROr:HOP: PATH:COUNT? <wsp>BERRor REI</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI. Selects the type of HOP (High Order Path) error. BERRor, selects the type of HOP (High Order Path) error as B3. REI, selects the REI-P (Remote Error Indicator - Path) error when bits 1 through 4 of the G1 byte contain one pattern from the following binary range: "#B0001" through "#B1000" (1 to 8).</p>
Response Syntax	<code><Count></code>

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:COUNT?

Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the count of High Order Path (HOP) error.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:MAN: TYPE BERR * SOUR:DATA:TEL:SON:ERR:HOP:PATH:AMO 15 * SOUR:DATA:TEL:SON:ERR:HOP:PATH:INJ * FETC:DATA:TEL:SON:ERR:HOP:PATH:COUN? BERR</pre> <p>Returns the error count.</p>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:AMount * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:INJect</pre>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERROr:
HOP:PATH:RATE?**

Description	<p>This query returns the current rate of High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SONet:ERROr:HOP: PATH:RATE?<wsp>BERRor REI</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor REI.</p> <p>Selects the type of HOP (High Order Path) error. BERRor, selects the type of HOP (High Order Path) error as B3.</p> <p>REI, selects the REI-P (Remote Error Indicator - Path) error when bits 1 through 4 of the G1 byte contain one pattern from the following binary range: "#B0001" through "#B1000" (1 to 8).</p>
Response Syntax	<code><Rate></code>

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor: HOP:PATH:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of High Order Path (HOP) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:MAN:TYPE BERR</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:AMO 15</p> <p>* SOUR:DATA:TEL:SON:ERR:HOP:PATH:INJ</p> <p>* FETC:DATA:TEL:SON:ERR:HOP:PATH:RATE? BERR</p> <p>Returns the error rate.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:AMOUNT</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:HOP:PATH:INJect</p>

:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:HISTory?

Description	<p>This query returns the history status of High Order Path (HOP) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:HISTory? <wsp>AIS LOP LOM RDI TIM PLM UNEQP1 PDI EPSD1 EPCD1 EPPD1</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>AIS LOP LOM RDI TIM PLM UNEQP1 PDI EPSD1 EPCD1 EPPD1.</pre> <p>Selects the type of HOP (High Order Path) alarm. AIS, selects the AIS-P (Alarm Indication Signal - Path) alarm when the H1 and H2 bytes for an Synchronous Transport Signal (STS) path contain an all-ones pattern in three consecutive frames or more.</p>

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: HOP:PATH:HISTory?

LOP, selects the LOP-P (Loss of Pointer - Path), for non-concatenated payloads, the LOP alarm indicates that a valid pointer is not found in N consecutive frames (where $8 \leq N \leq 10$), or N consecutive NDFs ("1001" pattern) are detected.

LOM, selects the LOM (Loss of Multi(where $8 \leq N \leq 10$)frame), for Virtual Tributary (VT) structured SONET frames, the LOM alarm indicates that the system loss track of the H4 byte multiframe indicator sequence.

RDI, selects the RDI-P (Remote Defect Indication - Path) when bits 5, 6 and 7 of the G1 byte contain the "100" or "111" pattern in five consecutive frames.

TIM, selects the TIM-P (Trace Identifier Mismatch - Path) that none of the sampled path trace strings match the expected message value. The TIM-P alarm result is only available when TIM-P from J1 Trace section has been enabled.

PLM, selects the PLM-P (Payload Label Mismatch - Path) declared upon receipt of five consecutive frames with mismatched STS signal labels.

UNEQP1, selects the UNEQP-P (Unequipped - Path) when the C2 bytes contain "00 H" in five consecutive frames.

**:FETCh[1..n]:DATA:TELecom:SONet:ALARm:
HOP:PATH:HISTory?**

PDI, selects the PDI-P (Payload Defect Indication - Path) when detecting Loss Of Pointer - VT (LOP-V), Alarm Indication Signal - VT (AIS-V), Digital Signal-level 3 (DS3) Alarm Indication Signal (AIS), Digital Signal-level 3 (DS3) Loss of Signal (LOS), or Digital Signal-level 3 (DS3) Out of Frame (OOF) defect on any Virtual Tributary (VT) or Digital Signal-level 3 (DS3) payload that it embeds into the STS SPE that it is originating.

EPSD1, selects the ERDI-PSD (Enhanced RDI - Path Server Defect) when bits 5, 6 and 7 of the G1 byte contain the "101" pattern in five consecutive frames.

EPCD1, selects the ERDI-PCD (Enhanced RDI - Path Connectivity Defect) alarm when bits 5, 6 and 7 of the G1 byte contain the "110" pattern in five consecutive frames.

EPPD1, selects the ERDI-PPD (Enhanced RDI - Path Payload Defect) alarm is declared when bits 5, 6 and 7 of the G1 byte contain the "010" pattern in five consecutive frames.

Response Syntax <History>

**:FETCh[1..n]:DATA:TELecom:SONet:ALARm:
HOP:PATH:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of High Order Path (HOP) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:HOP:PATH ON* FETC:DATA:TEL:SON:ALAR:HOP:PATH:HIST? AIS <p>Returns the alarm history.</p>
Note	<p>For 8120NGE/8130NGE/8130NGEv2 modules, choices are AIS LOP LOM RDI TIM PLM UNEQP1 PDI EPSD1 EPCD1 EPPD1.</p> <p>For 8140 module, choices are AIS LOP RDI TIM PLM UNEQP1 PDI EPSD1 EPCD1 EPPD1.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELecom:SONet:ALARm:HOP:PATH:TYPE* SOURce[1..n]:DATA:TELecom:SONet:ALARm:HOP:PATH

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:SEConds?**

Description	<p>This query returns the number of seconds within which High Order Path (HOP) alarm occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:SEConds? <wsp>AIS LOP LOM RDI TIM PLM UNEQP1 PDI EPSD1 EPCD1 EPPD1</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>AIS LOP LOM RDI TIM PLM UNEQP1 PDI EPSD1 EPCD1 EPPD1.</pre> <p>Selects the type of HOP (High Order Path) alarm. AIS, selects the AIS-P (Alarm Indication Signal - Path) alarm when the H1 and H2 bytes for an Synchronous Transport Signal (STS) path contain an all-ones pattern in three consecutive frames or more.</p>

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: HOP:PATH:SEConds?

LOP, selects the LOP-P (Loss of Pointer - Path), for non-concatenated payloads, the LOP alarm indicates that a valid pointer is not found in N consecutive frames (where $8 \leq N \leq 10$), or N consecutive NDFs ("1001" pattern) are detected. LOM, selects the LOM (Loss of Multiframe), for VT structured SONET frames, the LOM alarm indicates that the system loss track of the H4 byte multiframe indicator sequence.

RDI, selects the RDI-P (Remote Defect Indication - Path) when bits 5, 6 and 7 of the G1 byte contain the "100" or "111" pattern in five consecutive frames.

TIM, selects the TIM-P (Trace Identifier Mismatch - Path) that none of the sampled path trace strings match the expected message value. The TIM-P alarm result is only available when TIM-P from J1 Trace section has been enabled.

PLM, selects the PLM-P (Payload Label Mismatch - Path) declared upon receipt of five consecutive frames with mismatched STS signal labels.

UNEQP1, selects the UNEQP-P (Unequipped - Path) when the C2 bytes contain "00 H" in five consecutive frames.

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: HOP:PATH:SEConds?

PDI, selects the PDI-P (Payload Defect Indication - Path) when detecting Loss Of Pointer - VT (LOP-V), Alarm Indication Signal - VT (AIS-V), Digital Signal-level 3 (DS3) Alarm Indication Signal (AIS), Digital Signal-level 3 (DS3) Loss of Signal (LOS), or Digital Signal-level 3 (DS3) Out of Frame (OOF) defect on any Virtual Tributary (VT) or Digital Signal-level 3 (DS3) payload that it embeds into the STS SPE that it is originating.

EPSD1, selects the ERDI-PSD (Enhanced RDI - Path Server Defect) when bits 5, 6 and 7 of the G1 byte contain the "101" pattern in five consecutive frames.

EPCD1, selects the ERDI-PCD (Enhanced RDI - Path Connectivity Defect) alarm when bits 5, 6 and 7 of the G1 byte contain the "110" pattern in five consecutive frames.

EPPD1, selects the ERDI-PPD (Enhanced RDI - Path Payload Defect) alarm is declared when bits 5, 6 and 7 of the G1 byte contain the "010" pattern in five consecutive frames.

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of High Order Path (HOP) alarm.

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: HOP:PATH:SEConds?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:HOP:PATH ON* FETC:DATA:TEL:SON:ALAR:HOP:PATH:SEC? AIS Returns the number of seconds of HOP alarm.
Note	For 8120NGE/8130NGE/8130NGEv2 modules, choices are AIS LOP LOM RDI TIM PLM UNEQP1 PDI EPSD1 EPCD1 EPPD1. For 8140 module, choices are AIS LOP RDI TIM PLM UNEQP1 PDI EPSD1 EPCD1 EPPD1.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELecom:SONet:ALARm:HOP:PATH:TYPE* SOURce[1..n]:DATA:TELecom:SONet:ALARm:HOP:PATH

:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:CURRent?

Description	<p>This query returns the current status of High Order Path (HOP) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: HOP:PATH:CURRent?<wsp>AIS LOP LOM RDI TIM PLM UNEQP1 PDI EPSD1 EPCD1 EPPD1</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>AIS LOP LOM RDI TIM PLM UNEQP1 PDI EPSD1 EPCD1 EPPD1.</pre> <p>Selects the type of HOP (High Order Path) alarm.</p> <p>AIS, selects the AIS-P (Alarm Indication Signal - Path) alarm when the H1 and H2 bytes for an Synchronous Transport Signal (STS) path contain an all-ones pattern in three consecutive frames or more.</p>

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: HOP:PATH:CURRent?

LOP, selects the LOP-P (Loss of Pointer - Path), for non-concatenated payloads, the LOP alarm indicates that a valid pointer is not found in N consecutive frames (where $8 \leq N \leq 10$), or N consecutive NDFs ("1001" pattern) are detected.

LOM, selects the LOM (Loss of Multiframe), for Virtual Tributary (VT) structured SONET frames, the LOM (Loss of Multiframe) alarm indicates that the system loss track of the H4 byte multiframe indicator sequence.

RDI, selects the RDI-P (Remote Defect Indication - Path) when bits 5, 6 and 7 of the G1 byte contain the "100" or "111" pattern in five consecutive frames.

TIM, selects the TIM-P (Trace Identifier Mismatch - Path) that none of the sampled path trace strings match the expected message value. The TIM-P alarm result is only available when TIM-P from J1 Trace section has been enabled.

PLM, selects the PLM-P (Payload Label Mismatch - Path) declared upon receipt of five consecutive frames with mismatched Synchronous Transport Signal (STS) signal labels.

UNEQ1, selects the UNEQ-P (Unequipped - Path) when the C2 bytes contain "00 H" in five consecutive frames.

**:FETCh[1..n]:DATA:TELecom:SONet:ALARm:
HOP:PATH:CURRENT?**

PDI, selects the PDI-P (Payload Defect Indication - Path) when detecting Loss Of Pointer - VT (LOP-V), Alarm Indication Signal - VT (AIS-V), Digital Signal-level 3 (DS3) Alarm Indication Signal (AIS), Digital Signal-level 3 (DS3) Loss of Signal (LOS), or Digital Signal-level 3 (DS3) Out of Frame (OOF) defect on any Virtual Tributary (VT) or Digital Signal-level 3 (DS3) payload that it embeds into the STS SPE that it is originating.

EPSD1, selects the ERDI-PSD (Enhanced RDI - Path Server Defect) when bits 5, 6 and 7 of the G1 byte contain the "101" pattern in five consecutive frames.

EPCD1, selects the ERDI-PCD (Enhanced RDI - Path Connectivity Defect) alarm when bits 5, 6 and 7 of the G1 byte contain the "110" pattern in five consecutive frames.

EPPD1, selects the ERDI-PPD (Enhanced RDI - Path Payload Defect) alarm is declared when bits 5, 6 and 7 of the G1 byte contain the "010" pattern in five consecutive frames.

Response Syntax <Current>

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:CURRENT?**

Response(s)

Current:

The response data syntax for <Current> 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.

Example(s)

* SOUR:DATA:TEL:SON:ALAR:HOP:PATH:TYPE
AIS

* SOUR:DATA:TEL:SON:ALAR:HOP:PATH ON

* FETC:DATA:TEL:SON:ALAR:HOP:PATH:CURR?
AIS

Returns the current alarm status.

Note

For **8120NGE/8130NGE/8130NGEv2** modules, choices are AIS|LOP|LOM|RDI|TIM|PLM|UNEQP1|PDI|EPSD1|EPCD1|EPPD1.

For **8140** module, choices are AIS|LOP|RDI|TIM|PLM|UNEQP1|PDI|EPSD1|EPCD1|EPPD1.

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH:TYPE

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
HOP:PATH

:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:TYPE

Description	<p>This command selects the type of Low Order Path (LOP) alarm.</p> <p>At *RST, this value is set to AIS.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:TYPE<wsp>AIS RDI EVSD EVCD EVPD RFIV LOP UNEQv</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>AIS RDI EVSD EVCD EVPD RFIV LOP UNEQv.</p> <p>Selects the type of LOP (Low Order Path) alarm.</p> <p>AIS, selects the AIS-V (Alarm Indication Signal - VT) which generates all-ones pattern for the V1 and V2 bytes of the Virtual Tributary (VT) path and payload.</p> <p>RDI, selects the RDI-V (Remote Defect Indication - V) which generates "#B1" for the bit 8 of the V5 byte and a "#B00" pattern for bits 6 and 7 of Z7 byte.</p> <p>EVSD, selects the ERDI-VSD (Enhanced RDI - VT Server Defect) which generates "#B101" pattern for bits 5, 6, and 7 of the Z7 byte, and "1" for bit 8 of V5 byte.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:TYPE

EVCD, selects the ERDI-VCD (Enhanced RDI - VT Connectivity Defect) which generates "#B110" pattern for bits 5, 6, and 7 of Z7 byte, and "1" for bit 8 of V5 byte.

EVPD, selects the ERDI-VPD (Enhanced RDI - VT Payload Defect) which generates "#B010" pattern for bits 5, 6, and 7 of Z7 byte, and "0" for bit 8 of the V5 byte.

RFIV, selects the RFI-V (Remote Failure Indication - VT) alarm when bit 4 of the V5 byte contains "#B1" in five consecutive superframes.

LOP, selects the LOP-V (Loss of Pointer - VT) alarm which indicates that a valid pointer is not found in N consecutive superframes, or if N consecutive NDFs ("#B1001" pattern) are detected.

UNEQv, selects the UNEQ-V (Unequipped - VT) which generates samples of unequipped VT signal label (bits 5 through 7 of V5 byte are set to "#B000").

Example(s)

* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:TYPE
AIS
* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:TYPE?
Returns AIS

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:TYPE?

**:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:TYPE?**

Description	<p>This query returns the type of Low Order Path (LOP) alarm.</p> <p>At *RST, this value is set to AIS.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Low Order Path (LOP) alarm. AIS, Alarm Indication Signal - VT (AIS-V) is selected as Low Order Path (LOP) alarm. RDI, Remote Defect Indication - VT (RDI-V) is selected as Low Order Path (LOP) alarm. EVSD, Enhanced Remote Defect Indication - VT Server Defect (ERDI-VSD) is selected as Low Order Path (LOP) alarm. EVCD, Enhanced Remote Defect Indication - VT Connectivity Defect (ERDI-VCD) is selected as Low Order Path (LOP) alarm.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:TYPE?

EVPD, Enhanced Remote Defect Indication - VT Payload Defect (ERDI-VPD) is selected as Low Order Path (LOP) alarm.

RFIV, Remote Failure Indication - VT (RFI-V) is selected as Low Order Path (LOP) alarm.

LOP, Loss Of Pointer - VT (LOP-V) is selected as Low Order Path (LOP) alarm.

UNEQV, Unequipped - VT (UNEQ-V) is selected as Low Order Path (LOP) alarm.

Example(s)

* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:TYPE
AIS

* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:TYPE?
Returns AIS

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:TYPE

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH**

Description	<p>This command enables or disables the Low Order Path (LOP) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the LOP (Low Order Path) alarm generation.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:LOP:PATH ON* SOUR:DATA:TEL:SON:ALAR:LOP:PATH? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH?

**:SOURce[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH?**

Description	<p>This query returns the status of Low Order Path (LOP) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of Low Order Path (LOP) alarm generation.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ALAR:LOP:PATH:TYPE AIS* SOUR:DATA:TEL:SON:ALAR:LOP:PATH ON* SOUR:DATA:TEL:SON:ALAR:LOP:PATH? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LOP:PATH:MANual:TYPE**

Description	<p>This command selects the manual type of Low Order Path (LOP) error.</p> <p>At *RST, this value is set to BIP2.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LOP:PATH:MANual:TYPE<wsp>BIP2 REI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 REI.</p> <p>Selects the type of LOP (Low Order Path) error. BIP2, selects the BIP-2 (Bit-Interleaved Parity - 2 bits) error which indicates a parity error by performing a routine even-parity check over all VT1.5 bytes of the previous frame of a composite signal (VT1.5/VT2/VT6).</p> <p>REI, selects the REI-V (Remote Error Indication - VT) error when bit 3 of the V5 byte is set to "1".</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:MANual:TYPE**

Example(s) * SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN:
TYPE BIP2
* SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN:
TYPE? Returns BIP2

Note FTB/IQS-8140 Transport Blazer does not support
this command.

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:MANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LOP:PATH:MANual:TYPE?**

Description	<p>This query returns the manual type of Low Order Path (LOP) error.</p> <p>At *RST, this value is set to BIP2.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LOP:PATH:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Low Order Path (LOP) error. BIP2, Bit-Interleaved Parity - 2 bits (BIP2) is selected as Low Order Path (LOP) error. REI, Remote Error Indication - VT (REIV) is selected as Low Order Path (LOP) error.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:MANual:TYPE?

Example(s)	* SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN: TYPE BIP2 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN: TYPE? Returns BIP2
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:SONet:ERROr:
LOP:PATH:AMOUNT**

Description	<p>This command sets the amount of Low Order Path (LOP) error to be injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERROr: LOP:PATH:AMOUNT <wsp> <Amount> MAXimum MINimum</pre>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the amount of LOP (Low Order Path) error to be injected.</p> <p>Choices are 1 through 50.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AMOut**

Example(s) * SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN:
 TYPE BIP2
 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AMO 15
 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AMO?
 Returns 15

Note FTB/IQS-8140 Transport Blazer does not support
 this command.

See Also * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
 LOP:PATH:MANual:TYPE
 * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
 LOP:PATH:AMOut?

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:AMOUnt?**

Description	<p>This query returns the amount of Low Order Path (LOP) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AMOUnt? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<pre><Amount></pre>

:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AMOut?

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Low Order Path (LOP) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN:TYPE BIP2</p> <p>* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AMO 15</p> <p>* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AMO?</p> <p>Returns 15</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AMOut</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:ERRor:
LOP:PATH:INJect**

Description	<p>This command injects the type of Low Order Path (LOP) error.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN: TYPE BIP2 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AMO 15 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:INJ
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AMOut

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:
PATH:HISTory?**

Description	<p>This query returns the history status of Low Order Path (LOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP: PATH:HISTory? <wsp>BIP2 REI</p>
Parameter(s)	<p>Error: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 REI. Selects the type of LOP (Low Order Path) error. BIP2, selects the BIP-2 (Bit-Interleaved Parity - 2) bits error by performing a routine even-parity check over all VT1.5 bytes of the previous frame of a composite signal (VT1.5/VT2/VT6). REI, selects the REI-V (Remote Error Indication - VT) error when bit 3 of the V5 byte is set to "1".</p>
Response Syntax	<p><History></p>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:
PATH:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Low Order Path (LOP) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN:TYPE BIP2* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AMO 15* SOUR:DATA:TEL:SON:ERR:LOP:PATH:INJ* FETC:DATA:TEL:SON:ERR:LOP:PATH:HIST? BIP2 <p>Returns the error history status.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AMOUNT* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:
PATH:SEConds?**

Description This query returns the number of seconds within which Low Order Path (LOP) error occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:
PATH:SEConds? <wsp>BIP2|REI

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2|REI.
Selects the type of LOP (Low Order Path) error. BIP2, selects the BIP-2 (Bit-Interleaved Parity - 2) bits error by performing a routine even-parity check over all VT1.5 bytes of the previous frame of a composite signal (VT1.5/VT2/VT6). REI, selects the REI-V (Remote Error Indication - VT) error when bit 3 of the V5 byte is set to "1".

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:
PATH:SECOnds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Low Order Path (LOP) error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN:TYPE BIP2 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AMO 15 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:INJ * FETC:DATA:TEL:SON:ERR:LOP:PATH:SEC? BIP2 Returns the number of errored seconds.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AMOUNT * SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:INJECT

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:
PATH:CURRent?**

Description	<p>This query returns the current status of Low Order Path (LOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP: PATH:CURRent? <wsp>BIP2 REI</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 REI.</p> <p>Selects the type of LOP (Low Order Path) error. BIP2, selects the BIP-2 (Bit-Interleaved Parity - 2) bits error by performing a routine even-parity check over all VT1.5 bytes of the previous frame of a composite signal (VT1.5/VT2/VT6).</p> <p>REI, selects the REI-V (Remote Error Indication - VT) error when bit 3 of the V5 byte is set to "1".</p>
Response Syntax	<p><Current></p>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:
PATH:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of Low Order Path (LOP) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN:TYPE BIP2* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AMO 15* SOUR:DATA:TEL:SON:ERR:LOP:PATH:INJ* FETC:DATA:TEL:SON:ERR:LOP:PATH:CURR? BIP2 Returns the current error status.
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AMOUNT* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:INJect

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP: PATH:COUNT?

Description	<p>This query returns the count of Low Order Path (LOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP: PATH:COUNT? <wsp>BIP2 REI</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 REI.</p> <p>Selects the type of LOP (Low Order Path) error. BIP2, selects the BIP-2 (Bit-Interleaved Parity - 2) bits error by performing a routine even-parity check over all VT1.5 bytes of the previous frame of a composite signal (VT1.5/VT2/VT6).</p> <p>REI, selects the REI-V (Remote Error Indication - VT) error when bit 3 of the V5 byte is set to "1".</p>
Response Syntax	<p><Count></p>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:
PATH:COUNT?**

Response(s)	Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of Low Order Path (LOP) error.
Example(s)	* SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN: TYPE BIP2 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:AMO 15 * SOUR:DATA:TEL:SON:ERR:LOP:PATH:INJ * FETC:DATA:TEL:SON:ERR:LOP:PATH:COUN? BIP2 Returns the error count.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:AMOUNT * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: LOP:PATH:INJect

:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP: PATH:RATE?

Description This query returns the current rate of Low Order Path (LOP) error.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:
PATH:RATE? <wsp>BIP2|REI

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2|REI.
Selects the type of LOP (Low Order Path) error.
BIP2, selects the BIP-2 (Bit-Interleaved Parity - 2) bits error by performing a routine even-parity check over all VT1.5 bytes of the previous frame of a composite signal (VT1.5/VT2/VT6).
REI, selects the REI-V (Remote Error Indication - VT) error when bit 3 of the V5 byte is set to "1".

Response Syntax <Rate>

**:FETCh[1..n]:DATA:TELEcom:SONet:ERRor:LOP:
PATH:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of Low Order Path (LOP) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SON:ERR:LOP:PATH:MAN:TYPE BIP2* SOUR:DATA:TEL:SON:ERR:LOP:PATH:AMO 15* SOUR:DATA:TEL:SON:ERR:LOP:PATH:INJ* FETC:DATA:TEL:SON:ERR:LOP:PATH:RATE? BIP2 Returns the current error rate.
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:AMOUNT* SOURce[1..n]:DATA:TELEcom:SONet:ERRor:LOP:PATH:INJect

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:HISTory?**

Description	<p>This query returns the history status of Low Order Path (LOP) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:HISTory? <wsp> AIS LOP RDI RFIV TIM PLM UNEQv EVSD EVCD EVPD</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS LOP RDI RFIV TIM PLM UNEQv EVSD EVCD EVPD.</p> <p>Selects the type of LOP (Low Order Path) alarm.</p> <p>AIS, selects the AIS-V (Alarm Indication Signal - VT) when V1 and V2 bytes for the VT (Virtual Tributary) path contain an all-ones pattern in three consecutive superframes.</p> <p>LOP, selects the LOP-V (Loss Of Pointer - VT) alarm which indicates that a valid pointer is not found in N consecutive superframes (where 8 <= N <= 10), or if N consecutive NDFs ("1001" pattern) are detected.</p>

**:FETCh[1..n]:DATA:TELecom:SONet:ALARm:
LOP:PATH:HISTory?**

RDI, selects the RDI-V (Remote Defect Indication - VT) alarm when bit 8 of the V5 byte contains "1" in five consecutive VT (Virtual Tributary) superframes while bits 6 and 7 of the Z7 byte contain the "00" or "11" pattern.

RFIV, selects the RFI-V (Remote Failure Indication - VT) alarm when bit 4 of the V5 byte contains "1" in five consecutive superframes.

TIM, selects the TIM-V (Trace Identifier Mismatch - VT) which indicates that none of the sampled VT (Virtual Tributary) trace strings match the expected message value. The TIM-V alarm result is only available when TIM-V from J2 Trace section has been enabled.

PLM, selects the PLM-V (Payload Label Mismatch - VT) upon receipt of five consecutive superframes with mismatched VT Signal (bits 5 through 7 of the V5 byte are "000", "001" or "111").

UNEQv, selects the UNEQ-V (Unequipped - VT) when bit 5 through 7 of the V5 byte contain "000" for five consecutive superframes.

EVSD, selects the ERDI-VSD (Enhanced RDI - VT Server Defect) alarm when bits 5, 6, and 7 of the Z7 byte contain the "101" pattern, and bit 8 of the V5 byte contain "1", in five consecutive VT (Virtual Tributary) superframes.

EVCD, ERDI-VCD (Enhanced RDI - VT Connectivity Defect) alarm is declared when bits 5, 6, and 7 of the Z7 byte contain the "110" pattern, and bit 8 of the V5 byte contain "1", in five consecutive VT superframes.

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: LOP:PATH:HISTOrY?

EVPD, selects the ERDI-VPD (Enhanced RDI - VT Path Payload Defect) alarm is declared when bits 5, 6, and 7 of the Z7 byte contain the "#B010" pattern, and bit 8 of the V5 byte contain "0", in five consecutive VT (Virtual Tributary) superframes.

Response Syntax <History>

Response(s) History:
The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the history status Low Order Path (LOP) 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.

**:FETCh[1..n]:DATA:TELecom:SONet:ALARm:
LOP:PATH:HISTory?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE AIS* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH ON* FETC:DATA:TEL:SON:ALAR:LOP:PATH:HIST? AIS Returns the alarm history status.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELecom:SONet:ALARm:LOP:PATH:TYPE* SOURce[1..n]:DATA:TELecom:SONet:ALARm:LOP:PATH

**:FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:SEConds?**

Description This query returns the number of seconds within which Low Order Path (LOP) alarm occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SONet:ALARm:
LOP:PATH:SEConds? <wsp>AIS|LOP|RDI|RFIV|
TIM|PLM|UNEQv|EVSD|EVCD|EVPD

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
AIS|LOP|RDI|RFIV|TIM|PLM|UNEQv|EVSD|EVCD|EVPD.
Selects the type of LOP (Low Order Path) alarm.
AIS, selects the AIS-V (Alarm Indication Signal - VT) when V1 and V2 bytes for the VT (Virtual Tributary) path contain an all-ones pattern in three consecutive superframes.
LOP, selects the LOP-V (Loss Of Pointer - VT) alarm which indicates that a valid pointer is not found in N consecutive superframes (where 8 <= N <= 10), or if N consecutive NDFs ("1001" pattern) are detected.

**:FETCh[1..n]:DATA:TELecom:SONet:ALARm:
LOP:PATH:SEConds?**

RDI, selects the RDI-V (Remote Defect Indication - VT) alarm when bit 8 of the V5 byte contains "1" in five consecutive VT (Virtual Tributary) superframes while bits 6 and 7 of the Z7 byte contain the "00" or "11" pattern.

RFIV, selects the RFI-V (Remote Failure Indication - VT) alarm when bit 4 of the V5 byte contains "1" in five consecutive superframes.

TIM, selects the TIM-V (Trace Identifier Mismatch - VT) which indicates that none of the sampled VT (Virtual Tributary) trace strings match the expected message value. The TIM-V alarm result is only available when TIM-V from J2 Trace section has been enabled.

PLM, selects the PLM-V (Payload Label Mismatch - VT) upon receipt of five consecutive superframes with mismatched VT Signal (bits 5 through 7 of the V5 byte are "000", "001" or "111").

UNEQv, selects the UNEQ-V (Unequipped - VT) when bit 5 through 7 of the V5 byte contain "000" for five consecutive superframes.

EVSD, selects the ERDI-VSD (Enhanced RDI - VT Server Defect) alarm when bits 5, 6, and 7 of the Z7 byte contain the "101" pattern, and bit 8 of the V5 byte contain "1", in five consecutive VT (Virtual Tributary) superframes.

EVCD, ERDI-VCD (Enhanced RDI - VT Connectivity Defect) alarm is declared when bits 5, 6, and 7 of the Z7 byte contain the "110" pattern, and bit 8 of the V5 byte contain "1", in five consecutive VT (Virtual Tributary) superframes.

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: LOP:PATH:SEConds?

EVPD, selects the ERDI-VPD (Enhanced RDI - VT Path Payload Defect) alarm is declared when bits 5, 6, and 7 of the Z7 byte contain the "#B010" pattern, and bit 8 of the V5 byte contain "0", in five consecutive VT (Virtual Tributary) superframes.

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Low Order Path (LOP) alarm.

Example(s)

- * SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE AIS
- * SOUR:DATA:TEL:SDH:ALAR:LOP:PATH ON
- * FETC:DATA:TEL:SON:ALAR:LOP:PATH:SEC? AIS
returns the number of seconds of LOP alarm.

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELecom:SONet:ALARm:LOP:PATH:TYPE
- * SOURce[1..n]:DATA:TELecom:SONet:ALARm:LOP:PATH

:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:CURRent?

Description	<p>This query returns the current status of Low Order Path (LOP) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:ALARm: LOP:PATH:CURRent? <wsp>AIS LOP RDI RFIV TIM PLM UNEQv EVSD EVCD EVPD</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>AIS LOP RDI RFIV TIM PLM UNEQv EVSD EVCD EVPD.</pre> <p>Selects the type of LOP (Low Order Path) alarm.</p> <p>AIS, selects the AIS-V (Alarm Indication Signal - VT) when V1 and V2 bytes for the VT (Virtual Tributary) path contain an all-ones pattern in three consecutive superframes.</p> <p>LOP, selects the LOP-V (Loss Of Pointer - VT) alarm which indicates that a valid pointer is not found in N consecutive superframes (where $8 \leq N \leq 10$), or if N consecutive NDFs ("1001" pattern) are detected.</p>

**:FETCh[1..n]:DATA:TELecom:SONet:ALARm:
LOP:PATH:CURRent?**

RDI, selects the RDI-V (Remote Defect Indication - VT) alarm when bit 8 of the V5 byte contains "1" in five consecutive VT (Virtual Tributary) superframes while bits 6 and 7 of the Z7 byte contain the "00" or "11" pattern.

RFIV, selects the RFI-V (Remote Failure Indication - VT) alarm when bit 4 of the V5 byte contains "1" in five consecutive superframes.

TIM, selects the TIM-V (Trace Identifier Mismatch - VT) which indicates that none of the sampled VT trace strings match the expected message value. The TIM-V alarm result is only available when TIM-V from J2 Trace section has been enabled.

PLM, selects the PLM-V (Payload Label Mismatch - VT) upon receipt of five consecutive superframes with mismatched VT Signal (bits 5 through 7 of the V5 byte are "000", "001" or "111").

UNEQv, selects the UNEQ-V (Unequipped - VT) when bit 5 through 7 of the V5 byte contain "000" for five consecutive superframes.

EVSD, selects the ERDI-VSD (Enhanced RDI - VT Server Defect) alarm when bits 5, 6, and 7 of the Z7 byte contain the "101" pattern, and bit 8 of the V5 byte contain "1", in five consecutive VT (Virtual Tributary) superframes.

EVCD, ERDI-VCD (Enhanced RDI - VT Connectivity Defect) alarm is declared when bits 5, 6, and 7 of the Z7 byte contain the "110" pattern, and bit 8 of the V5 byte contain "1", in five consecutive VT (Virtual Tributary) superframes.

**:FETCh[1..n]:DATA:TELecom:SONet:ALARm:
LOP:PATH:CURREnt?**

EVPD, selects the ERDI-VPD (Enhanced RDI - VT Path Payload Defect) alarm is declared when bits 5, 6, and 7 of the Z7 byte contain the "#B010" pattern, and bit 8 of the V5 byte contain "0", in five consecutive VT (Virtual Tributary) superframes.

Response Syntax <Current>

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current status of Low Order Path (LOP) 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.

:FETCh[1..n]:DATA:TELecom:SONet:ALARm: LOP:PATH:CURRent?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE AIS* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH ON* FETC:DATA:TEL:SON:ALAR:LOP:PATH:CURR? AIS Returns the current alarm status.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELecom:SONet:ALARm: LOP:PATH:TYPE* SOURce[1..n]:DATA:TELecom:SONet:ALARm: LOP:PATH

:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERhead

Description	<p>This command sets the Section Overhead byte values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERhead <wsp> <Timeslot> ,A1 A2 J0 B1 E1 F1 D1 D2 D3 Z0 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33, <Value></pre>
Parameter(s)	<p>Timeslot: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the section overhead timeslot number.</p> <p>Overhead: The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: A1 A2 J0 B1 E1 F1 D1 D2 D3 Z0 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33. Selects the section overhead bytes. A1, selects F6 as a hexadecimal value for A1. A2, selects 28 as a hexadecimal value for A2. J0, selects J0 trace.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERhead

B1, selects B1 as a Bit Interleaved Parity code (BIP-8).

E1, selects E1 as an Orderwire.

F1, selects F1 as an User.

D1, selects D1 as a Data Communications Channel (DCC).

D2, selects D2 as a Data Communications Channel (DCC).

D3, selects D3 as a Data Communications Channel (DCC).

Z0, selects Z0 as a Growth.

Byte is specified in two ways.

In first method standard names are used.

Ex: A1, A2.

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.

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:
SECTION:OVERhead****Value:**

The program data syntax for the third parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Selects the section overhead byte values in hexadecimal format.

Example(s)

* SOUR:DATA:TEL:SON:OH:SECT:OVER
1,A1,#HF6

* SOUR:DATA:TEL:SON:OH:SECT:OVER? 1,A1
Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:OH:
SECTION:OVERhead?

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:
SECTion:OVERhead?**

Description	<p>This query returns the Section Overhead byte values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTion:OVERhead? <wsp> <Timeslot>,A1 A2 J0 B1 E1 F1 D1 D2 D3 Z0 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33</pre>
Parameter(s)	<p>Timeslot: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the section overhead timeslot number.</p> <p>Overhead: The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: A1 A2 J0 B1 E1 F1 D1 D2 D3 Z0 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33. Selects the section overhead bytes. A1, selects F6 as a hexadecimal value for A1. A2, selects 28 as a hexadecimal value for A2. J0, selects J0 trace.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:
SECTION:OVERhead?**

B1, selects B1 as a Bit Interleaved Parity code (BIP-8).

E1, selects E1 as an Orderwire.

F1, selects F1 as an User.

D1, selects D1 as a Data Communications Channel (DCC).

D2, selects D2 as a Data Communications Channel (DCC).

D3, selects D3 as a Data Communications Channel (DCC).

Z0, selects Z0 as a Growth.

Byte is specified in two ways.

In first method standard names are used.

Ex: A1, A2.

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.

Response Syntax <Value>

:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTion:OVERhead?

Response(s)	Value: The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the section overhead byte value.
Example(s)	* SOUR:DATA:TEL:SON:OH:SECT:OVER 1,A1,#F6 * SOUR:DATA:TEL:SON:OH:SECT:OVER? 1,A1 Returns #F6
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTion:OVERhead

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:
SECTION:DISabled**

Description	<p>This command clears an Overwrite for all bytes of section overhead.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:DISabled
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SON:OH:SECT:DIS

:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERwrite:ENABLEd

Description

This command enables or disables the generation of the selected section byte.

At *RST, this value is set to OFF.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:SONet:OH:  
SECTION:OVERwrite:ENABLEd<wsp>  
<Timeslot>,A1|A2|J0|B1|E1|F1|D1|D2|D3|  
Z0|UD11|UD12|UD13|UD21|UD22|UD23|UD31  
|UD32|UD33,<Set>
```

:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERwrite:ENABLEd

Parameter(s)

Timeslot:

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the section overhead timeslot number.

Overhead:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

A1|A2|J0|B1|E1|F1|D1|D2|D3|Z0|UD11|UD12
|UD13|UD21|UD22|UD23|UD31|UD32|UD33.

Selects the section overhead bytes.

A1, selects F6 as a hexadecimal value for A1.

A2, selects 28 as a hexadecimal value for A2.

J0, selects J0 trace.

B1, selects B1 as a Bit Interleaved Parity code (BIP-8).

E1, selects E1 as an Orderwire.

F1, selects F1 as an User.

D1, selects D1 as a Data Communications Channel (DCC).

D2, selects D2 as a Data Communications Channel (DCC).

D3, selects D3 as a Data Communications Channel (DCC).

Z0, selects Z0 as a Growth.

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:
SECTION:OVERwrite:ENABLEd**

Byte is specified in two ways.

In first method standard names are used.

Ex: A1, A2.

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.

Set:

The program data syntax for the third parameter is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the generation of the selected section byte.

Example(s)

* SOUR:DATA:TEL:SON:OH:SECT:OVER:ENAB
1,A1,ON

* SOUR:DATA:TEL:SON:OH:SECT:OVER:ENAB?
1,A1 Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:OH:
SECTION:OVERhead

* SOURce[1..n]:DATA:TELEcom:SONet:OH:
SECTION:OVERwrite:ENABLEd?

:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERwrite:ENABLEd?

Description	<p>This query returns the status of generation of the selected section byte.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERwrite:ENABLEd? <wsp> <Timeslot> ,A1 A2 J0 B1 E1 F1 D1 D2 D3 Z0 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33</pre>
Parameter(s)	<p>Timeslot:</p> <p>The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the section overhead timeslot number.</p> <p>Overhead:</p> <p>The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>A1 A2 J0 B1 E1 F1 D1 D2 D3 Z0 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33.</pre> <p>Selects the section overhead bytes.</p> <p>A1, selects F6 as a hexadecimal value for A1.</p> <p>A2, selects 28 as a hexadecimal value for A2.</p>

:SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERwrite:ENABLEd?

J0, selects J0 trace.

B1, selects B1 as a Bit Interleaved Parity code (BIP-8).

E1, selects E1 as an Orderwire.

F1, selects F1 as an User.

D1, selects D1 as a Data Communications Channel (DCC).

D2, selects D2 as a Data Communications Channel (DCC).

D3, selects D3 as a Data Communications Channel (DCC).

Z0, selects Z0 as a Growth.

Byte is specified in two ways.

In first method standard names are used.

Ex: A1, A2.

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.

Response Syntax <Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:
SECTION:OVERwrite:ENABLEd?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of generation of the selected section byte.
Example(s)	* SOUR:DATA:TEL:SON:OH:SECT:OVER:ENAB 1,A1,ON * SOUR:DATA:TEL:SON:OH:SECT:OVER:ENAB? 1,A1 Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERhead * SOURce[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERwrite:ENABLEd

**:FETCh[1..n]:DATA:TELEcom:SONet:OH:
SECTION:OVERwrite:STATus?**

Description	<p>This query returns status of the section byte in any timeslot having the overwrite selected or not.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERwrite:STATus?</code>
Parameter(s)	None
Response Syntax	<code><Status></code>
Response(s)	<p>Status:</p> <p>The response data syntax for <code><Status></code> is defined as a <code><CHARACTER RESPONSE DATA></code> element. Returns the status of the overwrite for section overhead.</p> <p>PRESENT, indicates the presence of output signal at an optical port.</p> <p>ABSENT, indicates the absence of output signal at an optical port.</p>
Example(s)	<p>* FETC:DATA:TEL:SON:OH:SECT:OVER:STAT? Returns the status of overwrite for section overhead.</p>

:SENSe[1..n]:DATA:TELEcom:SONet:OH: SECTION:OVERhead?

Description	<p>This query returns the Section Overhead byte values in hexadecimal format for the receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SONet:OH:SECTION:OVERhead? <wsp><Timeslot> ,A1 A2 J0 B1 E1 F1 D1 D2 D3 Z0 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33</pre>
Parameter(s)	<p>Timeslot: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the section overhead timeslot number.</p> <p>Overhead: The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: A1 A2 J0 B1 E1 F1 D1 D2 D3 Z0 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33.</p>

**:SENSe[1..n]:DATA:TELEcom:SONet:OH:
SECTION:OVERhead?**

Selects the section overhead bytes for the receiver.

A1, selects F6 as a hexadecimal value for A1.

A2, selects 28 as a hexadecimal value for A2.

J0, selects J0 trace.

B1, selects B1 as a Bit Interleaved Parity code (BIP-8).

E1, selects E1 as an Orderwire.

F1, selects F1 as an User.

D1, selects D1 as a Data Communications Channel (DCC).

D2, selects D2 as a Data Communications Channel (DCC).

D3, selects D3 as a Data Communications Channel (DCC).

Z0, selects Z0 as a Growth.

Byte is specified in two ways.

In first method standard names are used.

Ex: A1, A2.

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.

Response Syntax <Value>

**:SENSe[1..n]:DATA:TELEcom:SONet:OH:
SECTion:OVERhead?**

Response(s)	Value: The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the section overhead byte value for the receiver.
Example(s)	* SENS:DATA:TEL:SON:OH:SECT:OVER? 1,A1 Returns the section overhead byte value for the receiver.
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:OH:SECTion:OVERhead?

:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE: OVERhead

Description	<p>This command sets the Line Overhead byte values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[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 S1 M0 M1 Z1 Z2 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63, <Value></pre>
Parameter(s)	<p>Timeslot: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the line overhead timeslot number.</p> <p>Overhead: The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: H1 H2 H3 B2 K1 K2 D4 D5 D6 D7 D8 D9 D10 D11 D12 S1 M0 M1 Z1 Z2 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63.</p>

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERhead**

Selects the line overhead bytes.

H1, selects H1 as a Pointer.

H2, selects H2 as a Pointer.

H3, selects H3 as a Pointer Action.

B2, selects B2 as a Bit Interleaved Parity code (BIP-8).

K1, selects K1 as a Automatic Protection Switching (APS).

K2, selects K2 as a Automatic Protection Switching (APS).

D4, selects D4 as a Data Communications Channel (DCC).

D5, selects D5 as a Data Communications Channel (DCC).

D6, selects D6 as a Data Communications Channel (DCC).

D7, selects D7 as a Data Communications Channel (DCC).

D8, selects D8 as a Data Communications Channel (DCC).

D9, selects D9 as a Data Communications Channel (DCC).

D10, selects D10 as a Data Communications Channel (DCC).

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERhead**

D11, selects D11 as a Data Communications Channel (DCC).

D12, selects D12 as a Data Communications Channel (DCC).

S1, selects S1 as a Synchronization status.

M0, selects M0 as a Remote Error Indicator - Line (REI-L).

M1, selects M1 as a Remote Error Indicator - Line (REI-L).

E2, selects E2 as an Orderwire.

Z1, selects Z1 as a Growth.

Z2, selects Z2 as a Growth.

Byte is specified in two ways.

In first method standard names are used.

Ex: H1, H2.

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.

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERhead****Value:**

The program data syntax for the third parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Selects the line overhead byte values in hexadecimal format.

Example(s)

* SOUR:DATA:TEL:SON:OH:LINE:OVER
1,D4,#H00

* SOUR:DATA:TEL:SON:OH:LINE:OVER? 1,D4
Returns #H00

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERhead?

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:
LINE:OVERhead?**

Description This query returns the Line Overhead byte values in hexadecimal format.

At *RST, this value is device dependent.

Syntax :SOURce[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 |S1 |M0 |M1 |Z1 |Z2 |E2 |UD11 |UD12 |UD13 |UD21 |UD22 |UD23 |UD31 |UD32 |UD33 |UD41 |UD42 |UD43 |UD51 |UD52 |UD53 |UD61 |UD62 |UD63

Parameter(s) Timeslot:
The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.
Selects the line overhead timeslot number.

Overhead:
The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
H1 |H2 |H3 |B2 |K1 |K2 |D4 |D5 |D6 |D7 |D8 |D9 |D10 |D11 |D12 |S1 |M0 |M1 |Z1 |Z2 |E2 |UD11 |UD12 |UD13 |UD21 |UD22 |UD23 |UD31 |UD32 |UD33 |UD41 |UD42 |UD43 |UD51 |UD52 |UD53 |UD61 |UD62 |UD63.

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:
LINE:OVERhead?**

Selects the line overhead bytes.

H1, selects H1 as a Pointer.

H2, selects H2 as a Pointer.

H3, selects H3 as a Pointer Action.

B2, selects B2 as a Bit Interleaved Parity code (BIP-8).

K1, selects K1 as a Automatic Protection Switching (APS).

K2, selects K2 as a Automatic Protection Switching (APS).

D4, selects D4 as a Data Communications Channel (DCC).

D5, selects D5 as a Data Communications Channel (DCC).

D6, selects D6 as a Data Communications Channel (DCC).

D7, selects D7 as a Data Communications Channel (DCC).

D8, selects D8 as a Data Communications Channel (DCC).

D9, selects D9 as a Data Communications Channel (DCC).

D10, selects D10 as a Data Communications Channel (DCC).

D11, selects D11 as a Data Communications Channel (DCC).

D12, selects D12 as a Data Communications Channel (DCC).

S1, selects S1 as a Synchronization status.

M0, selects M0 as a Remote Error Indicator - Line (REI-L).

M1, selects M1 as a Remote Error Indicator - Line (REI-L).

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:
LINE:OVERhead?**

Z1, selects Z1 as a Growth.
Z2, selects Z2 as a Growth.
E2, selects E2 as an Orderwire.
Byte is specified in two ways.
In first method standard names are used.
Ex: H1, H2.
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.

Response Syntax <Value>

Response(s) Value:
The response data syntax for <Value> is defined
as a <HEXADECIMAL NUMERIC RESPONSE
DATA> element.
Returns the line overhead byte value.

Example(s) * SOUR:DATA:TEL:SON:OH:LINE:OVER
1,D4,#H00
* SOUR:DATA:TEL:SON:OH:LINE:OVER? 1,D4
Returns #H00

See Also * SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERhead

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:
DISabled**

Description	<p>This command clears an Overwrite for all bytes of line overhead.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE: DISabled
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SON:OH:LINE:DIS

:SOURce[1..n]:DATA:TELEcom:SONet:OH: LINE:OVERwrite:ENABLEd

Description

This command enables or disables the generation of the selected line byte.

At *RST, this value is set to OFF.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:SONet:OH:  
LINE:OVERwrite:ENABLEd<wsp><Timeslot>,  
H1|H2|H3|B2|K1|K2|D4|D5|D6|D7|D8|D9|  
D10|D11|D12|S1|M0|M1|Z1|Z2|E2|UD11|UD12  
|UD13|UD21|UD22|UD23|UD31|UD32|UD33|  
UD41|UD42|UD43|UD51|UD52|UD53|UD61|  
UD62|UD63,<Set>
```

Parameter(s)

Timeslot:

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the line overhead timeslot number.

:SOURce[1..n]:DATA:TELEcom:SONet:OH: LINE:OVERwrite:ENABLEd

Overhead:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

H1|H2|H3|B2|K1|K2|D4|D5|D6|D7|D8|D9|
D10|D11|D12|S1|M0|M1|Z1|Z2|E2|UD11|UD12
|UD13|UD21|UD22|UD23|UD31|UD32|UD33|
UD41|UD42|UD43|UD51|UD52|UD53|UD61|
UD62|UD63.

Selects the line overhead bytes.

H1, selects H1 as a Pointer.

H2, selects H2 as a Pointer.

H3, selects H3 as a Pointer Action.

B2, selects B2 as a Bit Interleaved Parity code (BIP-8).

K1, selects K1 as a Automatic Protection Switching (APS).

K2, selects K2 as a Automatic Protection Switching (APS).

D4, selects D4 as a Data Communications Channel (DCC).

D5, selects D5 as a Data Communications Channel (DCC).

D6, selects D6 as a Data Communications Channel (DCC).

D7, selects D7 as a Data Communications Channel (DCC).

D8, selects D8 as a Data Communications Channel (DCC).

:SOURce[1..n]:DATA:TELEcom:SONet:OH: LINE:OVERwrite:ENABLEd

D9, selects D9 as a Data Communications Channel (DCC).

D10, selects D10 as a Data Communications Channel (DCC).

D11, selects D11 as a Data Communications Channel (DCC).

D12, selects D12 as a Data Communications Channel (DCC).

S1, selects S1 as a Synchronization status.

M0, selects M0 as a Remote Error Indicator - Line (REI-L).

M1, selects M1 as a Remote Error Indicator - Line (REI-L).

E2, selects E2 as an Orderwire.

Z1, selects Z1 as a Growth.

Z2, selects Z2 as a Growth.

E2, selects E2 as an Orderwire.

Byte is specified in two ways.

In first method standard names are used.

Ex: H1, H2.

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.

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:
LINE:OVERwrite:ENABLEd****Set:**

The program data syntax for the third parameter is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the generation of the selected line byte.

Example(s)

* SOUR:DATA:TEL:SON:OH:LINE:OVER:ENAB 1,
D4,ON
* SOUR:DATA:TEL:SON:OH:LINE:OVER:ENAB?
1,D4 Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERhead
* SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERwrite:ENABLEd?

:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERwrite:ENABLEd?

Description	<p>This query returns the status of generation of the selected line byte.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERwrite:ENABLEd?<wsp><Timeslot>,H1 H2 H3 B2 K1 K2 D4 D5 D6 D7 D8 D9 D10 D11 D12 S1 M0 M1 Z1 Z2 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63</pre>
Parameter(s)	<p>Timeslot: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the line overhead timeslot number.</p>

**:SOURCE[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERwrite:ENABLEd?**

Overhead:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

H1|H2|H3|B2|K1|K2|D4|D5|D6|D7|D8|D9|
D10|D11|D12|S1|M0|M1|Z1|Z2|E2|UD11|UD12
|UD13|UD21|UD22|UD23|UD31|UD32|UD33|
UD41|UD42|UD43|UD51|UD52|UD53|UD61|
UD62|UD63.

Selects the line overhead bytes.

H1, selects H1 as a Pointer.

H2, selects H2 as a Pointer.

H3, selects H3 as a Pointer Action.

B2, selects B2 as a Bit Interleaved Parity code (BIP-8).

K1, selects K1 as a Automatic Protection Switching (APS).

K2, selects K2 as a Automatic Protection Switching (APS).

D4, selects D4 as a Data Communications Channel (DCC).

D5, selects D5 as a Data Communications Channel (DCC).

D6, selects D6 as a Data Communications Channel (DCC).

D7, selects D7 as a Data Communications Channel (DCC).

D8, selects D8 as a Data Communications Channel (DCC).

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERwrite:ENABLEd?**

D9, selects D9 as a Data Communications Channel (DCC).
D10, selects D10 as a Data Communications Channel (DCC).
D11, selects D11 as a Data Communications Channel (DCC).
D12, selects D12 as a Data Communications Channel (DCC).
S1, selects S1 as a Synchronization status.
M0, selects M0 as a Remote Error Indicator - Line (REI-L).
M1, selects M1 as a Remote Error Indicator - Line (REI-L).
Z1, selects Z1 as a Growth.
Z2, selects Z2 as a Growth.
E2, selects E2 as an Orderwire.
Byte is specified in two ways.
In first method standard names are used.
Ex: H1, H2.
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.

Response Syntax <Set>

**:SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERwrite:ENABLEd?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of generation of the selected line byte.
Example(s)	* SOUR:DATA:TEL:SON:OH:LINE:OVER:ENAB 1,D4,ON * SOUR:DATA:TEL:SON:OH:LINE:OVER:ENAB? 1,D4 Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE: OVERhead * SOURce[1..n]:DATA:TELEcom:SONet:OH:LINE: OVERwrite:ENABLEd

**:FETCh[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERwrite:STATus?**

Description	<p>This query returns status of the line byte in any timeslot having the overwrite selected or not.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SONet:OH:LINE: OVERwrite:STATus?
Parameter(s)	None
Response Syntax	<Status>
Response(s)	<p>Status:</p> <p>The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> element. Returns the status of the overwrite for section overhead.</p> <p>PRESENT, indicates the presence of output signal at an optical port.</p> <p>ABSENT, indicates the absence of output signal at an optical port.</p>
Example(s)	<p>* FETC:DATA:TEL:SON:OH:LINE:OVER:STAT?</p> <p>Returns the status of overwrite for line overhead.</p>

:SENSe[1..n]:DATA:TELEcom:SONet:OH:LINE:OVERhead?

Description	<p>This query returns the Line Overhead byte values in hexadecimal format for the receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>: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 S1 M0 M1 Z1 Z2 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63</pre>
Parameter(s)	<p>Timeslot: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the line overhead timeslot number.</p> <p>Overhead: The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: H1 H2 H3 B2 K1 K2 D4 D5 D6 D7 D8 D9 D10 D11 D12 S1 M0 M1 Z1 Z2 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63.</p>

:SENSe[1..n]:DATA:TELEcom:SONet:OH:LINE: OVERhead?

Selects the line overhead bytes for the receiver.

H1, selects H1 as a Pointer.

H2, selects H2 as a Pointer.

H3, selects H3 as a Pointer Action.

B2, selects B2 as a Bit Interleaved Parity code (BIP-8).

K1, selects K1 as a Automatic Protection Switching (APS).

K2, selects K2 as a Automatic Protection Switching (APS).

D4, selects D4 as a Data Communications Channel (DCC).

D5, selects D5 as a Data Communications Channel (DCC).

D6, selects D6 as a Data Communications Channel (DCC).

D7, selects D7 as a Data Communications Channel (DCC).

D8, selects D8 as a Data Communications Channel (DCC).

D9, selects D9 as a Data Communications Channel (DCC).

D10, selects D10 as a Data Communications Channel (DCC).

D11, selects D11 as a Data Communications Channel (DCC).

D12, selects D12 as a Data Communications Channel (DCC).

S1, selects S1 as a Synchronization status.

M0, selects M0 as a Remote Error Indicator - Line (REI-L).

**:SENSe[1..n]:DATA:TELEcom:SONet:OH:LINE:
OVERhead?**

M1, selects M1 as a Remote Error Indicator - Line (REI-L).

E2, selects E2 as an Orderwire.

Z1, selects Z1 as a Growth.

Z2, selects Z2 as a Growth.

Byte is specified in two ways.

In first method standard names are used.

Ex: H1, H2.

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.

Response Syntax <Value>

Response(s) Value:
The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.
Returns the line overhead byte value.

Example(s) * SENS:DATA:TEL:SON:OH:LINE:OVER? 1,H1
Returns the line overhead byte value for the receiver.

See Also * SOURce[1..n]:DATA:TELEcom:SONet:OH:
LINE:OVERhead

**:FETCh[1..n]:DATA:TELEcom:SONet:SECTion:
PM:STATistics?**

Description	<p>This query returns the performance monitoring statistics of Section.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SONet:SECTion:PM :STATistics? <wsp>G829ISM,EFS EB ES SES BBE UAS ESR SESR BBER,NEND</p>
Parameter(s)	<p>Standard:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: G829ISM.</p> <p>Selects the performance monitoring standard number.</p> <p>G829ISM, selects G.829 ISM as a standard number.</p>

**:FETCh[1..n]:DATA:TELEcom:SONet:SECTION:
PM:STATistics?**

Type:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

EFS|EB|ES|SES|BBE|UAS|ESR|SESR|BBER.

Selects the performance monitoring statistics.

EFS, selects Error Free Seconds (EFS).

EB, selects Errored Block (EB).

ES, selects Errored Seconds (ES).

SES, selects Severely Errored Seconds (SES).

BBE, selects Background Block Error (SES).

UAS, selects Unavailable Second (UAS).

ESR, selects Errored Second Ratio (ESR).

SESR, selects Severely Errored Second Ratio (SESR).

BBER, selects Background Block Error Ratio (BBER).

**:FETCh[1..n]:DATA:TELEcom:SONet:SECTion:
PM:STATistics?**

End:

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> element for this parameter is: NEND.

Selects the Near-End.

NEND, selects the standard for Near-End.

Response Syntax <Statistics>

Response(s) Statistics:

The response data syntax for <Statistics> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the performance monitoring statistics of Section.

Example(s) * FETC:DATA:TEL:SON:SECT:PM:STAT?
G829ISM,EFS,NEND Returns the performance monitoring statistics of Section.

**:FETCh[1..n]:DATA:TELEcom:SONet:LINE:PM:
STATistics?**

Description	<p>This query returns the performance monitoring statistics of Line.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SONet:LINE:PM: STATistics?<wsp>G829ISM M2101ISM,EFS EB ES SES BBE UAS ESR SESR BBER,NEND FEND</pre>
Parameter(s)	<p>Standard:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: G829ISM M2101ISM.</p> <p>Selects the performance monitoring standard number.</p> <p>G829ISM, selects G.829 ISM as a standard number.</p> <p>M2101ISM, selects M.2101 ISM as a standard number.</p>

:FETCh[1..n]:DATA:TELEcom:SONet:LINE:PM: STATistics?

Type:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

EFS|EB|ES|SES|BBE|UAS|ESR|SESR|BBER.

Selects the performance monitoring statistics.

EFS, selects Error Free Seconds (EFS).

EB, selects Errored Block (EB).

ES, selects Errored Seconds (ES).

SES, selects Severely Errored Seconds (SES).

BBE, selects Background Block Error (SES).

UAS, selects Unavailable Second (UAS).

ESR, selects Errored Second Ratio (ESR).

SESR, selects Severely Errored Second Ratio (SESR).

BBER, selects Background Block Error Ratio (BBER).

:FETCh[1..n]:DATA:TELEcom:SONet:LINE:PM:STATistics?

End:

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NEND|FEND.

Selects Near-End or Far-End.

NEND, selects the standard for Near-End.

FEND, selects the standard for Far-End.

Response Syntax

<Statistics>

Response(s)

Statistics:

The response data syntax for <Statistics> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the performance monitoring statistics of Line.

Example(s)

```
* FETC:DATA:TEL:SON:LINE:PM:STAT?
G829ISM,EFS,NEND
```

Returns the performance monitoring statistics of Line.

SDH/SONET Common Command Reference

:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:ENABLEd

Description	<p>This command enables or disables the trace. When enabled, generates the J1/J2 Trace message defined to give access to the trace format and message. When the J1/J2 Trace is disabled, the J1/J2 1-byte format is used.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:ENABLEd<wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the J1/J2 trace.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON * SOUR:DATA:TEL:SDHS:OVER:J1:ENAB? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:ENABLEd?</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:ENABLEd?**

Description	This query returns the status of J1/J2 trace. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns 0 if J1/J2 trace is disabled, or returns 1.
Example(s)	* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON * SOUR:DATA:TEL:SDHS:OVER:J1:ENAB? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:ENABLEd

:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:OVERwrite:ENABLEd

Description This command enables or disables the Overwrite feature.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:OVERwrite:ENABLEd<wsp>
<Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Overwrite feature.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:OVERwrite:ENABLEd**

Example(s)	* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON * SOUR:DATA:TEL:SDHS:OVER:J1:OVER:ENAB ON * SOUR:DATA:TEL:SDHS:OVER:J1:OVER:ENAB? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J1:ENABLEd * SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J1:OVERwrite:ENABLEd?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:OVERwrite:ENABLEd?**

Description	This query returns the status of the Overwrite feature. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:OVERwrite:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Overwrite feature.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:OVERwrite:ENABLEd?****Example(s)**

* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON
* SOUR:DATA:TEL:SDHS:OVER:J1:OVER:ENAB
ON
* SOUR:DATA:TEL:SDHS:OVER:J1:OVER:ENAB?
Returns 1

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J1:ENABled
* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J1:OVERwrite:ENABled

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern**

Description This command sets the J1/J2 trace value in 16 or 64 bytes format.

At *RST, this value is set to B16.

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern<wsp>B16|B64

Parameter(s) Pattern:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: B16|B64.
Sets the format for J1/J2 trace.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATTern**

B16, selects the 16 Byte format.

B64, selects the 64 Byte format.

Example(s)

* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON

* SOUR:DATA:TEL:SDHS:OVER:J1:PATT B16

* SOUR:DATA:TEL:SDHS:OVER:J1:PATT?

Returns B16

See Also

* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:ENABLEd

* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATTern?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern?**

Description	This query returns the J1/J2 trace value in 16 or 64 bytes format. At *RST, this value is set to B16.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern?
Parameter(s)	None
Response Syntax	<Pattern>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATTern?**

Response(s)	<p>Pattern:</p> <p>The response data syntax for <Pattern> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the format for J1/J2 trace.</p> <p>B16, 16 Byte format is selected.</p> <p>B64, 64 Byte format is selected.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON</p> <p>* SOUR:DATA:TEL:SDHS:OVER:J1:PATT B16</p> <p>* SOUR:DATA:TEL:SDHS:OVER:J1:PATT?</p> <p>Returns B16</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:ENABLEd</p> <p>* SOURce[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:PATTern</p>

:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern:B16

Description

This command sets the 16 bytes format of the J1/J2 trace.

At *RST, this value is set to "EXFO SONET/SDH".

Syntax

:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern:B16<wsp>
<Message>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern:B16**

Parameter(s)	Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the 16 bytes format for J1/J2 trace.
Example(s)	* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON * SOUR:DATA:TEL:SDHS:OVER:J1:PATT B16 * SOUR:DATA:TEL:SDHS:OVER:J1:PATT:B16 "EXFO SONET/SDH" * SOUR:DATA:TEL:SDHS:OVER:J1:PATT:B16? Returns "EXFO SONET/SDH"
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:ENABLEd * SOURce[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:PATtern * SOURce[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:PATtern:B16?

:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern:B16?

Description	This query returns the 16 bytes format of the J1/J2 trace. At *RST, this value is set to "EXFO SONET/SDH".
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern:B16?
Parameter(s)	None
Response Syntax	<Message>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern:B16?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the 16 bytes format for J1/J2 trace.

Example(s)

* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON

* SOUR:DATA:TEL:SDHS:OVER:J1:PATT B16

* SOUR:DATA:TEL:SDHS:OVER:J1:PATT:B16
"EXFO SONET/SDH"

* SOUR:DATA:TEL:SDHS:OVER:J1:PATT:B16?
Returns "EXFO SONET/SDH"

See Also

* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:ENABLEd

* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern

* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern:B16

:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern:B64

Description

This command sets the 64 bytes format of the J1/J2 trace.

At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".

Syntax

:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern:B64<wsp>
<Message>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern:B64****Parameter(s)**

Message:

The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.

Sets the 64 byte format for J1/J2 trace.

Example(s)

* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON
* SOUR:DATA:TEL:SDHS:OVER:J1:PATT B64
* SOUR:DATA:TEL:SDHS:OVER:J1:PATT:B64
"EXFO SONET/SDH Analyzer Section/RS trace test message"
* SOUR:DATA:TEL:SDHS:OVER:J1:PATT:B64?
Returns "EXFO SONET/SDH Analyzer Section/RS trace test message"

See Also

* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:ENABLEd
* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern
* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern:B64?

:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern:B64?

Description	<p>This query returns the 64 bytes format of the J1/J2 trace.</p> <p>At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:PATtern:B64?</code>
Parameter(s)	None
Response Syntax	<Message>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern:B64?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the 64 bytes format for J1/J2 trace.

Example(s)

* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON
* SOUR:DATA:TEL:SDHS:OVER:J1:PATT B64
* SOUR:DATA:TEL:SDHS:OVER:J1:PATT:B64
"EXFO SONET/SDH Analyzer Section/RS trace test message"
* SOUR:DATA:TEL:SDHS:OVER:J1:PATT:B64?
Returns "EXFO SONET/SDH Analyzer Section/RS trace test message"

See Also

* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:ENABLEd
* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern
* SOURce[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:PATtern:B64

:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM

Description	<p>This command enables or disables the Trace Identifier Mismatch (TIM) for the expected message defined.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the TIM (Trace Identifier Mismatch).</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDHS:OVER:J1:TIM ON* SENS:DATA:TEL:SDHS:OVER:J1:TIM? Returns 1
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM?

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM?**

Description	<p>This query returns the Trace Identifier Mismatch (TIM) for the expected message defined.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Trace Identifier Mismatch (TIM).</p>
Example(s)	<p>* SENS:DATA:TEL:SDHS:OVER:J1:TIM ON</p> <p>* SENS:DATA:TEL:SDHS:OVER:J1:TIM? Returns 1</p>
See Also	* SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM

:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern

Description	<p>This command sets the format expected in the J1/J2 trace.</p> <p>At *RST, this value is set to B16.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern<wsp>B16 B64</pre>
Parameter(s)	<p>Pattern:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: B16 B64.</p> <p>Sets the format expected in the J1/J2 trace.</p>

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern**

B16, selects the 16 Byte format.

B64, selects the 64 Byte format.

Example(s)

* SENS:DATA:TEL:SDHS:OVER:J1:TIM ON

* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT B16

* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT?

Returns B16

See Also

* SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM

* SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern?

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern?**

Description This query returns the format expected in the J1/J2 trace.

At *RST, this value is set to B16.

Syntax :SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern?

Parameter(s) None

Response Syntax <Pattern>

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern?**

Response(s)	<p>Pattern:</p> <p>The response data syntax for <Pattern> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the format expected in the J1/J2 trace.</p> <p>B16, 16 Byte format is selected.</p> <p>B64, 64 Byte format is selected.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDHS:OVER:J1:TIM ON* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT B16* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT? <p>Returns B16</p>
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:TIM* SENSe[1..n]:DATA:TELEcom:SDHSonet:OVERhead:J[1..n]:TIM:PATtern

:FETCh[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:RECEived?

Description This query returns the received J1/J2 value in 16 or 64 bytes format.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern:RECEived?

Parameter(s) None

Response Syntax <Message>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern:RECeived?**

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the received J1/J2 value in 16 or 64 bytes format.
Example(s)	* SOUR:DATA:TEL:SDHS:OVER:J1:ENAB ON * SOUR:DATA:TEL:SDHS:OVER:J1:PATT:B64 "EXFO SONET/SDH Analyzer Section/RS trace test message" * FETC:DATA:TEL:SDHS:OVER:J1:TIM:PAT:REC? Returns the J1/J2 value in 16 or 64 bytes format.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J1:ENABled * SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J1:PATtern:B16 * SOURce[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J1:PATtern:B64

:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B16

Description

This command sets the expected message for 16 bytes format of J1/J2 trace.

At *RST, this value is set to "EXFO SONET/SDH".

Syntax

```
:SENSe[1..n]:DATA:TELEcom:SDHSonet:  
OVERhead:J[1..n]:TIM:PATtern:B16<wsp>  
<Message>
```

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern:B16**

Parameter(s)	Message: The program data syntax for the parameter is defined as a <STRING RESPONSE DATA> element. Sets the expected message for 16 bytes format.
Example(s)	* SENS:DATA:TEL:SDHS:OVER:J1:TIM ON * SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT B16 * SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT:B16 "EXFO SONET/SDH" * SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT:B16? Returns "EXFO SONET/SDH"
See Also	* SENSE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM * SENSE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern * SENSE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B16?

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern:B16?**

Description	This query returns the expected message for 16 bytes format of J1/J2 trace. At *RST, this value is set to "EXFO SONET/SDH".
Syntax	:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B16?
Parameter(s)	None
Response Syntax	<Message>

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern:B16?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the expected message for 16 bytes format.

Example(s)

```
* SENS:DATA:TEL:SDHS:OVER:J1:TIM ON
* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT B16
* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT:B16
"EXFO SONET/SDH"
* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT:B16?
Returns "EXFO SONET/SDH"
```

See Also

```
* SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM
* SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern
* SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern:B16
```

:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B64

Description

This command sets the expected message for 64 bytes format of J1/J2 trace.

At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".

Syntax

:SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern:B64<wsp>
<Message>

:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B64

Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the expected message for 64 bytes format.</p>
Example(s)	<p>* SENS:DATA:TEL:SDHS:OVER:J1:TIM ON</p> <p>* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT B64</p> <p>* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT:B64 "EXFO SONET/SDH Analyzer Section/RS trace test message"</p> <p>* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT:B64? Returns "EXFO SONET/SDH Analyzer Section/RS trace test message"</p>
See Also	<p>* SENSE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM</p> <p>* SENSE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern</p> <p>* SENSE[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B64?</p>

:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B64?

Description	<p>This query returns the 64 bytes format of the J1/J2 trace of the expected message.</p> <p>At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SDHSonet: OVERhead:J[1..n]:TIM:PATtern:B64?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern:B64?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the expected message for 64 bytes format.

Example(s)

```
* SENS:DATA:TEL:SDHS:OVER:J1:TIM ON
* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT B64
* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT:B64
"EXFO SONET/SDH Analyzer Section/RS trace test
message"
* SENS:DATA:TEL:SDHS:OVER:J1:TIM:PATT:B64?
Returns "EXFO SONET/SDH Analyzer Section/RS
trace test message"
```

See Also

```
* SENSE[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM
* SENSE[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern
* SENSE[1..n]:DATA:TELEcom:SDHSonet:
OVERhead:J[1..n]:TIM:PATtern:B64
```

:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:VALue?

Description	<p>This query returns the current pointer value of High Order Path (HOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:POINter:VALue?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Size></p>
Response(s)	<p>Size: The response data syntax for <Size> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the current pointer value.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDHS:HOP:POIN:VAL? Returns the current pointer value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:INCRement:SIZE**

Description	<p>This command selects the number of positive pointer adjustment to be included into the SONET or SDH for High Order Path (HOP).</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:SIZE<wsp> <Size> MAXimum MINimum</pre>
Parameter(s)	<p>Size:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:INCRement:SIZE**

Sets the pointer size.
Choices are 1 through 1000.

Example(s)

* SOUR:DATA:TEL:SDHS:HOP:POIN:INCR:SIZE 15
* SOUR:DATA:TEL:SDHS:HOP:POIN:INCR:SIZE?
Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:INCRement:SIZE?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:INCRement:SIZE?**

Description	<p>This query returns the selected number of positive pointer adjustment to be included into the SONET or SDH.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:SIZE? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current number of positive pointer will be returned.</p>
Response Syntax	<pre><Size></pre>

:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:SIZE?

Response(s)	Size: The response data syntax for <Size> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the pointer size.
Example(s)	* SOUR:DATA:TEL:SDHS:HOP:POIN:INCR:SIZE 15 * SOUR:DATA:TEL:SDHS:HOP:POIN:INCR:SIZE? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:SIZE

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:DECREment:SIZE**

Description	<p>This command selects the number of negative pointer adjustment to be included into the SONET or SDH.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECREment:SIZE<wsp><Size> MAXimum MINimum</pre>
Parameter(s)	<p>Size:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:DECREment:SIZE**

Selects the pointer size.
Choices are 1 through 1000.

Example(s)

* SOUR:DATA:TEL:SDHS:HOP:POIN:DECR:SIZE
15
* SOUR:DATA:TEL:SDHS:HOP:POIN:DECR:SIZE?
Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:DECREment:SIZE?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:DECREment:SIZE?**

Description	<p>This query returns the selected number of negative pointer adjustment to be included into the SONET or SDH.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECREment:SIZE?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current number of negative pointer will be returned.</p>
Response Syntax	<pre><Size></pre>

:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement:SIZE?

Response(s)	<p>Size:</p> <p>The response data syntax for <Size> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the pointer size.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDHS:HOP:POIN:DECR:SIZE 15</p> <p>* SOUR:DATA:TEL:SDHS:HOP:POIN:DECR:SIZE? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement:SIZE</p>

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:INCRement**

Description	<p>This command sets 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>
Syntax	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SDHS:HOP:POIN:INCR:SIZE 15 * SOUR:DATA:TEL:SDHS:HOP:POIN:INCR
See Also	* SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:SIZE

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:DECRement**

Description	<p>This command sets 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>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement</p>
Parameter(s)	<p>None</p>
Example(s)	<p>* SOUR:DATA:TEL:SDHS:HOP:POIN:DECR:SIZE 15 * SOUR:DATA:TEL:SDHS:HOP:POIN:DECR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement:SIZE</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NEW:VALue**

Description	<p>This command sets the new pointer value for SONET/SDH of High Order Path (HOP).</p> <p>At *RST, this value reverts the default value.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:VALue<wsp> <Value> MAXimum MINimum</pre>
Parameter(s)	<p>Value:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NEW:VALue**

Selects the new pointer value between 0 through 782.

Example(s)

* SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:VAL 15
* SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:VAL?
Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NEW:VALue?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NEW:VALue?**

Description	<p>This query returns the new pointer value for High Order Path (HOP).</p> <p>At *RST, this value reverts the default value.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:VALue?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current value of new pointer will be returned.</p>
Response Syntax	<pre><Value></pre>

:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:VALue?

Response(s)	Value: The response data syntax for <Value> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the new pointer value.
Example(s)	* SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:VAL 15 * SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:VAL? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:VALue

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NEW**

Description	<p>This command 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>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:VAL 15 * SOUR:DATA:TEL:SDHS:HOP:POIN:NEW
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:VALue

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NEW:FLAG**

Description	<p>This command selects the status of a new pointer data flag for High Order Path (HOP).</p> <p>At *RST, this value is set to NNDF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:FLAG <wsp> NNDF NDF</p>
Parameter(s)	<p>Flag:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NNDF NDF.</p> <p>Selects the new pointer data flag.</p> <p>NNDF, selects the No New Data Flag (NNDF).</p> <p>NDF, selects the New Data Flag (NDF).</p>
Example(s)	<p>* SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:FLAG NNDF</p> <p>* SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:FLAG? Returns NNDF</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:FLAG?</p>

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NEW:FLAG?**

Description	<p>This query returns the status of a new pointer data flag for High Order Path (HOP).</p> <p>At *RST, this value is set to NNDF.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:FLAG?
Parameter(s)	None
Response Syntax	<Flag>
Response(s)	<p>Flag:</p> <p>The response data syntax for <Flag> is defined as a <CHARACTER RESPONSE DATA> element. 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>
Example(s)	<p>* SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:FLAG NNDF</p> <p>* SOUR:DATA:TEL:SDHS:HOP:POIN:NEW:FLAG? Returns NNDF</p>
See Also	* SOURCE[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:FLAG

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:VALue?**

Description	<p>This query returns the current pointer value received for High Order Path (HOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:VALue?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Pointer></p>
Response(s)	<p>Pointer: The response data syntax for <Pointer> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the actual pointer value being transmitted.</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:HOP:POIN:VAL? Returns the actual pointer value being transmitted.</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:OFFSet?**

Description	<p>This query returns the difference between the pointer increment and the pointer decrement for High Order Path (HOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:OFFSet?
Parameter(s)	None
Response Syntax	<Offset>
Response(s)	<p>Offset:</p> <p>The response data syntax for <Offset> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the difference between the pointer increment and the pointer decrement.</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:HOP:POIN:OFFS?</p> <p>Returns the actual pointer value being transmitted.</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:INCRement:COUNT?**

Description	<p>This query returns the count in which positive pointer adjustment is detected for High Order Path (HOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:COUNT?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Count></p>
Response(s)	<p>Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of positive pointer adjustment detected.</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:HOP:POIN:INCR:COUN? Returns the count of positive pointer adjustment detected.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:DECREment:COUNT?**

Description	This query returns the counts in which negative pointer adjustment is detected for High Order Path (HOP). At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECREment:COUNT?
Parameter(s)	None
Response Syntax	<Count>

:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement:COUNT?

Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the count of negative pointer adjustment detected.</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:HOP:POIN:DECR: COUN?</p> <p>Returns the count of negative pointer adjustment detected.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:INCRement:SEConds?**

Description	<p>This query returns the number of seconds in which positive pointer adjustment is detected for High Order Path (HOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement:SEConds?
Parameter(s)	None
Response Syntax	<Seconds>
Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of seconds of positive pointer adjustment detected.</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:HOP:POIN:INCR:SEC?</p> <p>Returns the number of seconds of positive pointer adjustment detected.</p>
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:INCRement

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:DECRement:SEConds?**

Description	<p>This query returns the number of seconds in which negative pointer adjustment is detected for High Order Path (HOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement:SEConds?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Seconds></p>
Response(s)	<p>Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of negative pointer adjustment detected.</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:HOP:POIN:DECR:SEC? Returns the number of seconds of negative pointer adjustment detected.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:DECRement</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NDF:COUNT?**

Description	<p>This query returns the count of New Data Flag (NDF).</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NDF:COUNT?
Parameter(s)	None
Response Syntax	<Count>
Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of count of New Data Flag (NDF).</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:HOP:POIN:NDF:COUN?</p> <p>Returns the number of counts of New Data Flag (NDF).</p>
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:FLAG

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NNDF:COUNT?**

Description	<p>This query returns the count of No New Data Flag (NNDF).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NNDF:COUNT?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Count></p>
Response(s)	<p>Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the number of counts of No New Data Flag (NNDF).</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:HOP:POIN:NNDF: COUN? Returns the number of counts of No New Data Flag (NNDF).</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:FLAG</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NDF:SEConds?**

Description	<p>This query returns the number of seconds of New Data Flag (NDF) for High Order Path (HOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NDF:SEConds?
Parameter(s)	None
Response Syntax	<Seconds>
Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of seconds of New Data Flag (NDF).</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:HOP:POIN:NDF:SEC?</p> <p>Returns the number of seconds of New Data Flag (NDF).</p>
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:FLAG

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
POINter:NNDF:SEConds?**

Description	<p>This query returns the number of seconds of No New Data Flag (NNDF) for High Order Path (HOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NNDF:SEConds?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Seconds></p>
Response(s)	<p>Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of No New Data Flag (NNDF).</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:HOP:POIN:NNDF:SEC? Returns the number of seconds of No New Data Flag (NNDF).</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDHSonet:HOP: POINter:NEW:FLAG</p>

:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:VALue?

Description	<p>This query returns the current pointer value being transmitted for Low Order Path (LOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:VALue?
Parameter(s)	None
Response Syntax	<Size>
Response(s)	<p>Size:</p> <p>The response data syntax for <Size> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current pointer value being transmitted.</p>
Example(s)	* SOUR:DATA:TEL:SDHS:LOP:POIN:VAL? Returns the current pointer value being transmitted.
Note	FTB/IQS-8140 Transport Blazer does not support this query.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:INCRement:SIZE**

Description	<p>This command selects the number of positive pointer adjustment to be included into the SONET or SDH for Low Order Path (LOP).</p> <p>At *RST, this value reverts the default value.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:SIZE<wsp> <Size> MAXimum MINimum</p>
Parameter(s)	<p>Size:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the pointer size.</p> <p>Choices are 1 through 1000.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:INCRement:SIZE**

Example(s)	* SOUR:DATA:TEL:SDHS:LOP:POIN:INCR:SIZE 15 * SOUR:DATA:TEL:SDHS:LOP:POIN:INCR:SIZE? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:INCRement:SIZE?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:INCRement:SIZE?**

Description	<p>This query returns the number of positive pointer adjustment to be included into the SONET or SDH for Low Order Path (LOP).</p> <p>At *RST, this value reverts the default value.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:SIZE?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current number of positive pointer will be returned.</p>
Response Syntax	<p><Size></p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:INCRement:SIZE?**

Response(s)	Size: The response data syntax for <Size> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the pointer size.
Example(s)	* SOUR:DATA:TEL:SDHS:LOP:POIN:INCR:SIZE 15 * SOUR:DATA:TEL:SDHS:LOP:POIN:INCR:SIZE? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:INCRement:SIZE

:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECREment:SIZE

Description	<p>This command selects the number of negative pointer adjustment to be included into the SONET or SDH.</p> <p>At *RST, this value reverts the default value.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECREment:SIZE<wsp><Size> MAXimum MINimum</pre>
Parameter(s)	<p>Size:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this Parameter are: MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p> <p>Selects the pointer size.</p> <p>Choices are 1 through 1000.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:DECRement:SIZE**

Example(s)	* SOUR:DATA:TEL:SDHS:LOP:POIN:DECR:SIZE 15 * SOUR:DATA:TEL:SDHS:LOP:POIN:DECR:SIZE? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement:SIZE?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:DECRement:SIZE?**

Description	<p>This query returns the number of negative pointer adjustment to be included into the SONET or SDH.</p> <p>At *RST, this value reverts the default value.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement:SIZE?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current number of negative pointer will be returned.</p>
Response Syntax	<p><Size></p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:DECRement:SIZE?**

Response(s)	Size: The response data syntax for <Size> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the pointer size.
Example(s)	* SOUR:DATA:TEL:SDHS:LOP:POIN:DECR:SIZE 15 * SOUR:DATA:TEL:SDHS:LOP:POIN:DECR:SIZE? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement:SIZE

:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement

Description	<p>This command sets the new positive pointer value for Low Order Path (LOP).</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement</code>
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDHS:LOP:POIN:INCR:SIZE 15* SOUR:DATA:TEL:SDHS:LOP:POIN:INCR
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:SIZE

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:DECRement**

Description	<p>This command sets the new negative pointer value for Low Order Path (LOP).</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SDHS:LOP:POIN:DECR:SIZE 15 * SOUR:DATA:TEL:SDHS:LOP:POIN:DECR
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement:SIZE

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:VALue**

Description This command selects the new pointer value for Low Order Path (LOP).

At *RST, this value reverts the default value.

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:VALue<wsp><Value>
|MAXimum|MINimum

Parameter(s) Value:
The program data syntax for the parameter is defined as a <numeric_value> element.
The allowed <numeric_value> elements for this parameter are: MAXimum|MINimum.
MAXimum allows to set the instrument to the greatest supported value.
MINimum allows to set the instrument to the smallest supported value.
Selects the new pointer value between 0 through 103.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:VALue**

Example(s)	* SOUR:DATA:TEL:SDHS:LOP:POIN:NEW:VAL 15 * SOUR:DATA:TEL:SDHS:LOP:POIN:NEW:VAL? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:NEW:VALue?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:VALue?**

Description	<p>This query returns the new pointer value.</p> <p>At *RST, this value reverts the default value.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW:VALue?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current number of new pointer value will be returned.</p>
Response Syntax	<p><Value></p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:VALue?**

Response(s)	Value: The response data syntax for <Value> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the new pointer value.
Example(s)	* SOUR:DATA:TEL:SDHS:LOP:POIN:NEW:VAL 15 * SOUR:DATA:TEL:SDHS:LOP:POIN:NEW:VAL? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:NEW:VALue

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW**

Description	This command sets the new pointer value. This command is an event and has no associated *RST condition or query form.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SDHS:LOP:POIN:NEW:VAL 15 * SOUR:DATA:TEL:SDHS:LOP:POIN:NEW
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW:VALue

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:FLAG**

Description	<p>This command selects the status of a new pointer data flag.</p> <p>At *RST, this value is set to NNDF.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW:FLAG <wsp> NNDF NDF</pre>
Parameter(s)	<p>Flag:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NNDF NDF.</p> <p>Selects the new pointer flag.</p> <p>NNDF, selects the No New Data Flag (NNDF).</p> <p>NDF, selects the New Data Flag (NDF).</p>

:SOURce[1..n]:DATA:TELecom:SDHSonet:LOP: POINter:NEW:FLAG

Example(s)	* SOUR:DATA:TEL:SDHS:LOP:POIN:NEW:FLAG NDF * SOUR:DATA:TEL:SDHS:LOP:POIN:NEW:FLAG? Returns NDF
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELecom:SDHSonet:LOP: POINter:NEW:FLAG?

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:FLAG?**

Description	<p>This query returns the status of a new pointer data flag.</p> <p>At *RST, this value is set to NNDF.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW:FLAG?
Parameter(s)	None
Response Syntax	<Flag>
Response(s)	<p>Flag:</p> <p>The response data syntax for <Flag> is defined as a <CHARACTER RESPONSE DATA> element. Returns the pointer flag.</p> <p>NNDF, No New Data Flag (NNDF) is selected.</p> <p>NDF, New Data Flag (NDF) is selected.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:FLAG?**

Example(s) * SOUR:DATA:TEL:SDHS:LOP:POIN:NEW:FLAG
NDF
* SOUR:DATA:TEL:SDHS:LOP:POIN:NEW:FLAG?
Returns NDF

Note FTB/IQS-8140 Transport Blazer does not support
this query.

See Also * SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:FLAG

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:VALue?**

Description	<p>This query returns the current pointer value received for Low Order Path (LOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:VALue?
Parameter(s)	None
Response Syntax	<Pointer>
Response(s)	<p>Pointer:</p> <p>The response data syntax for <Pointer> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the actual pointer value being transmitted.</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:LOP:POIN:VAL?</p> <p>Returns the actual pointer value being transmitted.</p>
Note	FTB/IQS-8140 Transport Blazer does not support this query.

:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:OFFSet?

Description	<p>This query returns the difference between the pointer increment and the pointer decrement.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:OFFSet?</code>
Parameter(s)	None
Response Syntax	<code><Offset></code>
Response(s)	<p>Offset:</p> <p>The response data syntax for <code><Offset></code> is defined as a <code><NR1 NUMERIC RESPONSE DATA></code> element.</p> <p>Returns the pointer offset value.</p>
Example(s)	<p>* FETC:DATA:TEL:SDHS:LOP:POIN:OFFS?</p> <p>Returns the pointer offset value.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:INCRement:COUNT?**

Description	<p>This query returns the count in which positive pointer adjustment is detected for Low Order Path (LOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:COUNT?
Parameter(s)	None
Response Syntax	<Count>
Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of count of positive pointer adjustment detected.</p>

:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:COUNT?

Example(s)	* FETC:DATA:TEL:SDHS:LOP:POIN:INCR:COUN? Returns the number of count of positive pointer adjustment detected.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:INCRement

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:DECREment:COUNT?**

Description	<p>This query returns the count in which negative pointer adjustment is detected.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECREment:COUNT?</pre>
Parameter(s)	None
Response Syntax	<Count>
Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of count of negative pointer adjustment detected.</p>

:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECRement:COUNT?

Example(s) * FETC:DATA:TEL:SDHS:LOP:POIN:DECR:COUN?
Returns the number of count of negative pointer adjustment detected.

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also * SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:DECRement

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:INCRement:SEConds?**

Description	<p>This query returns number of seconds in which positive pointer adjustment is detected.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:SEConds?
Parameter(s)	None
Response Syntax	<Seconds>
Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of seconds of positive pointer adjustment detected.</p>

:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:INCRement:SEConds?

Example(s)	* FETC:DATA:TEL:SDHS:LOP:POIN:INCR:SEC? Returns the number of count of positive pointer adjustment detected.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:INCRement

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:DECREment:SEConds?**

Description	<p>This query returns number of seconds in which negative pointer adjustment is detected.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:DECREment:SEConds?</code>
Parameter(s)	None
Response Syntax	<Seconds>
Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of seconds of negative pointer adjustment detected.</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:DECRement:SEConds?**

Example(s) * FETC:DATA:TEL:SDHS:LOP:POIN:DECR:SEC?
Returns the number of count of positive pointer adjustment detected.

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also * SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:POINter:DECRement

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NDF:COUNT?**

Description	<p>This query returns the count of New Data Flag (NDF) of Low Order Path (LOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NDF:COUNT?</code>
Parameter(s)	None
Response Syntax	<Count>
Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of counts of New Data Flag (NDF).</p>

:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NDF:COUNT?

Example(s)	* FETC:DATA:TEL:SDHS:LOP:POIN:NDF:COUN? Returns the number of count of New Data Flag (NDF).
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW:FLAG

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NNDF:COUNT?**

Description	<p>This query returns the count of No New Data Flag (NNDF) of High Order Path (HOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NNDF:COUNT?
Parameter(s)	None
Response Syntax	<Count>
Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of counts of No New Data Flag (NNDF).</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NNDF:COUNT?**

Example(s) * FETC:DATA:TEL:SDHS:LOP:POIN:NNDF:COUN?
Returns the number of count of No New Data
Flag (NNDF).

Note FTB/IQS-8140 Transport Blazer does not support
this query.

See Also * SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:FLAG

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NDF:SEConds?**

Description	<p>This query returns the number of seconds of New Data Flag (NDF) of Low Order Path (LOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NDF:SEConds?</code>
Parameter(s)	None
Response Syntax	<Seconds>
Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of seconds of New Data Flag (NDF).</p>

:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NDF:SEConds?

Example(s) * FETC:DATA:TEL:SDHS:LOP:POIN:NDF:SEC?
Returns the number of seconds of New Data Flag (NDF).

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also * SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NEW:FLAG

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
POINter:NNDF:SEConds?**

Description	<p>This query returns the number of seconds of No New Data Flag (NNDF) of High Order Path (HOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NNDF:SEConds?</code>
Parameter(s)	None
Response Syntax	<Seconds>
Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of seconds of No New Data Flag (NNDF).</p>

:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NNDF:SEConds?

Example(s)	* FETC:DATA:TEL:SDHS:LOP:POIN:NNDF:SEC? Returns the number of seconds of No New Data Flag (NDF).
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:LOP: POINter:NEW:FLAG

:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERhead

Description	<p>This command sets the High Order Path (HOP) overhead values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERhead<wsp>J1 B3 C2 G1 F2 H4 Z3 Z4 N1 F3 K3,<Value></pre>
Parameter(s)	<p>Overhead:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: J1 B3 C2 G1 F2 H4 Z3 Z4 N1 F3 K3.</p> <p>Selects the High Oder Path (HOP) overhead values.</p> <p>J1, selects the J1 trace. B3, selects Bit Interleaved Parity code (BIP-8). C2, selects the C2 as Path Signal Label. G1, selects G1 as a Path Status. F2, selects F2 as an User Channel. H4, selects H4 as a Multiframe Indicator. Z3, selects Z3 as a Growth. Z4, selects Z4 as a Growth. N1, selects N1 as a Tandem Connection Monitoring.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
HOP:OVERhead**

F3, selects F3 as an User Channel.
K3, selects K3 as a Automatic Protection
Switching (APS).

Value:

The program data syntax for the second
parameter is defined as a <NONDECIMAL
NUMERIC PROGRAM DATA> element.
Selects the HOP overhead values in hexadecimal
format in hexadecimal format.

Example(s)

* SOUR:DATA:TEL:SDHS:OH:HOP:OVER
C2,#H02
* SOUR:DATA:TEL:SDHS:OH:HOP:OVER? C2
Returns #H02

See Also

* SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
HOP:OVERhead?

:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERhead?

Description	<p>This query returns the High Order Path (HOP) overhead values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERhead?<wsp>J1 B3 C2 G1 F2 H4 Z3 Z4 N1 F3 K3</pre>
Parameter(s)	<p>Overhead:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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, selects the J1 trace. B3, selects Bit Interleaved Parity code (BIP-8). C2, selects C2 as a Path Signal Label. G1, selects G1 as a Path Status. F2, selects F2 as an User Channel. H4, selects H4 as a Multiframe Indicator. Z3, selects Z3 as a Growth. Z4, selects Z4 as a Growth. N1, selects N1 as a Tandem Connection Monitoring.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
HOP:OVERhead?**

F3, selects F3 as an User Channel.
K3, selects K3 as a Automatic Protection
Switching (APS).

Response Syntax <Value>

Response(s) Value:
The response data syntax for <Value> is defined
as a <HEXADECIMAL NUMERIC RESPONSE
DATA> element.
Returns the High Order Path (HOP) overhead
byte value.

Example(s) * SOUR:DATA:TEL:SDHS:OH:HOP:OVER
C2,#H02
* SOUR:DATA:TEL:SDHS:OH:HOP:OVER? C2
Returns #H02

See Also * SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
HOP:OVERhead

:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERhead?

Description	<p>This query returns the High Order Path (HOP) overhead values in hexadecimal format for the receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERhead?<wsp>J1 B3 C2 G1 F2 H4 Z3 Z4 N1 F3 K3</pre>
Parameter(s)	<p>Overhead:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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, selects the J1 trace.</p> <p>B3, selects Bit Interleaved Parity code (BIP-8).</p> <p>C2, selects the C2 as Path Signal Label.</p> <p>G1, selects G1 as Path Status.</p> <p>F2, selects F2 as User Channel.</p> <p>H4, selects H4 as Multiframe Indicator.</p> <p>Z3, selects Z3 as Growth.</p> <p>Z4, selects Z4 as Growth.</p> <p>N1, selects N1 as Tandem Connection Monitoring.</p>

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

F3, selects F3 as User Channel.
K3, selects K3 as Automatic Protection Switching (APS).

Response Syntax <Value>

Response(s) Value:
The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.
Returns the High Order Path (HOP) overhead byte value for the receiver.

Example(s) * SENS:DATA:TEL:SDHS:OH:HOP:OVER? J1
Returns the HOP overhead byte value for the receiver.

See Also * SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
HOP:OVERhead?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
HOP:DISabled**

Description	<p>This command clears the Overwrite for all bytes of High Order Path (HOP).</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:DISabled
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SDHS:OH:HOP:DIS
Note	FTB/IQS-8140 Transport Blazer does not support this command.

:SOURCE[1..n]:DATA:TELEcom:SDHSonet: OH:HOP:OVERwrite:ENABLEd

Description

This command enables or disables the generation of the selected overhead byte for High Order Path (HOP).

At *RST, this value is set to OFF.

Syntax

:SOURCE[1..n]:DATA:TELEcom:SDHSonet:OH:
HOP:OVERwrite:ENABLEd<wsp>J1|B3|C2|G1|
F2|H4|Z3|Z4|N1|F3|K3,<Set>

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:
OH:HOP:OVERwrite:ENABLEd****Parameter(s)**

Overhead:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

J1|B3|C2|G1|F2|H4|Z3|Z4|N1|F3|K3.

Selects the High Order Path (HOP) overhead values.

J1, selects the J1 trace.

B3, selects Bit Interleaved Parity code (BIP-8).

C2, selects the C2 as Path Signal Label.

G1, selects G1 as Path Status.

F2, selects F2 as User Channel.

H4, selects H4 as Multiframe Indicator.

Z3, selects Z3 as Growth.

Z4, selects Z4 as Growth.

N1, selects N1 as Tandem Connection Monitoring.

F3, selects F3 as User Channel.

K3, selects K3 as Automatic Protection Switching (APS).

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OH:HOP:OVERwrite:ENABLEd**

Set:

The program data syntax for the second parameter is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the generation of the selected byte.

Example(s)

- * SOUR:DATA:TEL:SDHS:OH:HOP:OVER:ENAB C2,ON
- * SOUR:DATA:TEL:SDHS:OH:HOP:OVER:ENAB? C2 Returns 1

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERhead
- * SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:HOP:OVERwrite:ENABLEd?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OH:HOP:OVERwrite:ENABLEd?****Description**

This query returns the status for generation of the selected overhead byte for High Order Path (HOP).

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
HOP:OVERwrite:ENABLEd?<wsp>J1|B3|C2|G1|
F2|H4|Z3|Z4|N1|F3|K3

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OH:HOP:OVERwrite:ENABLEd?**

Parameter(s)	<p>Overhead:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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, selects the J1 trace.</p> <p>B3, selects Bit Interleaved Parity code (BIP-8).</p> <p>C2, selects the C2 as Path Signal Label.</p> <p>G1, selects G1 as Path Status.</p> <p>F2, selects F2 as User Channel.</p> <p>H4, selects H4 as Multiframe Indicator.</p> <p>Z3, selects Z3 as Growth.</p> <p>Z4, selects Z4 as Growth.</p> <p>N1, selects N1 as Tandem Connection Monitoring.</p> <p>F3, selects F3 as User Channel.</p> <p>K3, selects K3 as Automatic Protection Switching (APS).</p>
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
OH:HOP:OVERwrite:ENABLEd?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the overhead byte value for High Order Path (HOP).
Example(s)	* SOUR:DATA:TEL:SDHS:OH:HOP:OVER:ENAB C2,ON * SOUR:DATA:TEL:SDHS:OH:HOP:OVER:ENAB? C2 Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERhead * SOURce[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERwrite:ENABLEd

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
OH:HOP:OVERwrite:STATus?**

Description	This query returns the status for any byte in any timeslot having the Overwrite selected or not. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet:OH: HOP:OVERwrite:STATus?
Parameter(s)	None
Response Syntax	<Status>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
OH:HOP:OVERwrite:STATus?**

Response(s)	Status: The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> element. Returns the status for the selected byte. PRESENT, indicates the presence of output signal at an optical port. ABSENT, indicates the absence of output signal at an optical port.
Example(s)	* FETC:DATA:TEL:SDHS:OH:HOP:OVER:STAT? Returns the status for the selected byte.
Note	FTB/IQS-8140 Transport Blazer does not support this query.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
LOP:OVERhead**

Description	<p>This command sets the Low Order Path (LOP) overhead values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH: LOP:OVERhead <wsp> V5 J2 Z6 Z7 N2 K4, <Value></p>
Parameter(s)	<p>Overhead:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: V5 J2 Z6 Z7 N2 K4.</p> <p>Selects the Low Order Path (LOP) overhead byte.</p> <p>V5, selects V5 as a VT Path Overhead.</p> <p>J2, selects J2 as a VT Path Trace.</p> <p>Z6, selects as a VT Tandem Connection Monitoring.</p> <p>Z7, selects as an Extended Signal Label.</p> <p>N2, selects as a Network Operator Byte.</p> <p>K4, selects as a Extended Signal Label.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
LOP:OVERhead****Value:**

The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Selects the Low Order Path (LOP) overhead values in hexadecimal format.

Example(s)

* SOUR:DATA:TEL:SDHS:OH:LOP:OVER Z7, #H01
* SOUR:DATA:TEL:SDHS:OH:LOP:OVER? Z7
Returns #H01

See Also

* SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
LOP:OVERhead?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
LOP:OVERhead?**

Description	<p>This query returns the Low Order Path (LOP) overhead values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH: LOP:OVERhead? <wsp>V5 J2 Z6 Z7 N2 K4</p>
Parameter(s)	<p>Overhead:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: V5 J2 Z6 Z7 N2 K4.</p> <p>Selects the Low Order Path (LOP) overhead byte.</p> <p>V5, selects V5 as a VT Path Overhead.</p> <p>J2, selects J2 as a VT Path Trace.</p> <p>Z6, selects as a VT Tandem Connection Monitoring.</p> <p>Z7, selects as an Extended Signal Label.</p> <p>N2, selects as a Network Operator Byte.</p> <p>K4, selects as an Extended Signal Label.</p>
Response Syntax	<p><Value></p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:
LOP:OVERhead?**

Response(s)	Value: The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the Low Order Path (LOP) overhead byte value.
Example(s)	* SOUR:DATA:TEL:SDHS:OH:LOP:OVER Z7, #H01 * SOUR:DATA:TEL:SDHS:OH:LOP:OVER? Z7 Returns #H01
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:LOP:OVERhead

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH:
LOP:OVERhead?**

Description	<p>This query returns the Low Order Path (LOP) overhead values in hexadecimal format for the receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH: LOP:OVERhead? <wsp>V5 J2 Z6 Z7 N2 K4</p>
Parameter(s)	<p>Overhead: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: V5 J2 Z6 Z7 N2 K4. Selects the Low Order Path (LOP) overhead byte. V5, selects V5 as a VT Path Overhead. J2, selects J2 as a VT Path Trace. Z6, selects as a VT Tandem Connection Monitoring. Z7, selects as a Extended Signal Label. N2, selects as a Network Operator Byte. K4, selects as a Extended Signal Label.</p>
Response Syntax	<p><Value></p>

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:OH:
LOP:OVERhead?**

Response(s)	Value: The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the High Order Path (HOP) overhead byte value for the receiver.
Example(s)	* SENS:DATA:TEL:SDHS:OH:LOP:OVER? V5 Returns the HOP overhead byte value for the receiver.
See Also	* SOURce[1..n]:DATA:TELEcom:SDHSonet:OH:LOP:OVERhead?

:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:PM:STATistics?

Description This query returns the performance monitoring statistics of High Order Path (HOP).

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:PM:STATistics? <wsp>G828ISM|M2101ISM,EFS|EB|ES|SES|BBE|UAS|ESR|SESR|BBER|SEP|SEPI,NEND|FEND

Parameter(s) Standard:
The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
G828ISM|M2101ISM.
Selects the performance monitoring standard number.
G828ISM, selects G.828 ISM as a standard number.
M2101ISM, selects M.2101 ISM as a standard number.

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
PM:STATistics?**

Type:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

EFS|EB|ES|SES|BBE|UAS|ESR|SESR|BBER|
SEP|SEPI.

Selects the performance monitoring statistics.

EFS, selects Error Free Seconds (EFS).

EB, selects Errored Block (EB).

ES, selects Errored Seconds (ES).

SES, selects Severely Errored Seconds (SES).

BBE, selects Background Block Error (SES).

UAS, selects Unavailable Second (UAS).

ESR, selects Errored Second Ratio (ESR).

SESR, selects Severely Errored Second Ratio (SESR).

BBER, selects Background Block Error Ratio (BBER).

SEP, selects Severely Errored Period (SEP).

SEPI, selects Severely Errored Period Intensity (SEPI).

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:HOP:
PM:STATistics?**

End:

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

NEND|FEND.

Selects Near-End or Far-End.

NEND, selects the standard for Near-End.

FEND, selects the standard for Far-End.

Response Syntax <Statistics>

Response(s) Statistics:

The response data syntax for <Statistics> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the performance monitoring statistics of High Order Path (HOP).

Example(s) * FETC:DATA:TEL:SDHS:HOP:PM:STAT?
G828ISM, EFS, NEND

Returns the performance monitoring statistics of HOP.

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:
PM:STATistics?**

Description	<p>This query returns the performance monitoring statistics of Low Order Path (LOP).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP:PM :STATistics? <wsp>G828ISM M2101ISM,EFS EB ES SES BBE UAS ESR SESR BBER SEP SEPI, NEND FEND</pre>
Parameter(s)	<p>Standard:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: G828ISM M2101ISM.</p> <p>Selects the performance monitoring standard number.</p> <p>G828ISM, selects G.828 ISM as a standard number.</p> <p>M2101ISM, selects M.2101 ISM as a standard number.</p>

:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: PM:STATistics?

Type:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

EFS|EB|ES|SES|BBE|UAS|ESR|SESR|BBER|
SEP|SEPI.

Selects the performance monitoring statistics.

EFS, selects Error Free Seconds (EFS).

EB, selects Errored Block (EB).

ES, selects Errored Seconds (ES).

SES, selects Severely Errored Seconds (SES).

BBE, selects Background Block Error (SES).

UAS, selects Unavailable Second (UAS).

ESR, selects Errored Second Ratio (ESR).

SESR, selects Severely Errored Second Ratio (SESR).

BBER, selects Background Block Error Ratio (BBER).

SEP, selects Severely Errored Period (SEP).

SEPI, selects Severely Errored Period Intensity (SEPI).

:FETCh[1..n]:DATA:TELEcom:SDHSonet:LOP: PM:STATistics?

End:

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

NEND | FEND.

Selects Near-End or Far-End.

NEND, selects the standard for Near-End.

FEND, selects the standard for Far-End.

Response Syntax

<Statistics>

Response(s)

Statistics:

The response data syntax for <Statistics> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the performance monitoring statistics of Low Order Path (LOP).

Example(s)

```
* FETC:DATA:TEL:SDHS:LOP:PM:STAT?
G828ISM, EFS, NEND
```

Returns the performance monitoring statistics of LOP.

SDH Command Reference

:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:ENABLEd

Description This command enables or disables the J0 trace. When enabled, generates the J0 Trace message defined to give access to the trace format and message. When the J0 Trace is disabled, the J0 1-byte format is used.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:
J:ENABLEd<wsp><Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the J0 trace.

Example(s) * SOUR:DATA:TEL:SDH:OVER:J:ENAB ON
* SOUR:DATA:TEL:SDH:OVER:J:ENAB? Returns 1

See Also * SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:
J:ENABLEd?

**:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:
J:ENABled?**

Description	This query returns the status of the J0 trace. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:ENABled?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns 0 if J0 trace is disabled, or returns 1.
Example(s)	* SOUR:DATA:TEL:SDH:OVER:J:ENAB ON * SOUR:DATA:TEL:SDH:OVER:J:ENAB? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:ENABled

**:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead
:J:OVERwrite:ENABled**

Description	<p>This command enables or disables the Overwrite feature.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J :OVERwrite:ENABled <wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Overwrite feature.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead
:J:OVERwrite:ENABLEd**

Example(s)	* SOUR:DATA:TEL:SDH:OVER:J:ENAB ON * SOUR:DATA:TEL:SDH:OVER:J:OVER:ENAB ON * SOUR:DATA:TEL:SDH:OVER:J:OVER:ENAB? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:ENABLEd * SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:OVERwrite:ENABLEd?

:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead :J:OVERwrite:ENABLEd?

Description	This query returns the status of the Overwrite feature. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead :J:OVERwrite:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Overwrite feature.

**:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead
:J:OVERwrite:ENABLEd?**

Example(s)	* SOUR:DATA:TEL:SDH:OVER:J:ENAB ON * SOUR:DATA:TEL:SDH:OVER:J:OVER:ENAB ON * SOUR:DATA:TEL:SDH:OVER:J:OVER:ENAB? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:ENABLEd * SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:OVERwrite:ENABLEd

:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern

Description	<p>This command sets the J0 trace in 16 or 64 bytes format.</p> <p>At *RST, this value is set to B16.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern<wsp>B16 B64</p>
Parameter(s)	<p>Pattern:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: B16 B64.</p> <p>Sets the format for J0 trace.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:
J:PATtern**

B16, selects the 16 Bytes format.

B64, selects the 64 Bytes format.

Example(s)

* SOUR:DATA:TEL:SDH:OVER:J:ENAB ON

* SOUR:DATA:TEL:SDH:OVER:J:PATT B16

* SOUR:DATA:TEL:SDH:OVER:J:PATT?

Returns B16

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:
J:ENABled

* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:
J:PATtern?

:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern?

Description	This query returns the J0 trace in 16 or 64 bytes format. At *RST, this value is set to B16.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern?
Parameter(s)	None
Response Syntax	<Pattern>

**:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:
J:PATtern?**

Response(s)	Pattern: The response data syntax for <Pattern> is defined as a <CHARACTER RESPONSE DATA> element. Returns the format for J0 trace. B16, 16 Bytes format is selected. B64, 64 Bytes format is selected.
Example(s)	* SOUR:DATA:TEL:SDH:OVER:J:ENAB ON * SOUR:DATA:TEL:SDH:OVER:J:PATT B16 * SOUR:DATA:TEL:SDH:OVER:J:PATT? Returns B16
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:ENABled * SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:PATTern

:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern:B16

Description	<p>This command sets the 16 bytes format of the J0 trace.</p> <p>At *RST, this value is set to "EXFO SONET/SDH".</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern:B16<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the 16 bytes format for J0 trace.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SDH:OVER:J:ENAB ON * SOUR:DATA:TEL:SDH:OVER:J:PATT B16 * SOUR:DATA:TEL:SDH:OVER:J:PATT:B16 "EXFO SONET/SDH" * SOUR:DATA:TEL:SDH:OVER:J:PATT:B16? Returns "EXFO SONET/SDH"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:ENABled * SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern * SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern:B16?</pre>

**:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J
:PATtern:B16?**

Description	This query returns the 16 bytes format of the J0 trace. At *RST, this value is set to "EXFO SONET/SDH".
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern:B16?
Parameter(s)	None
Response Syntax	<Message>

:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J :PATtern:B16?

Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the 16 bytes format for J0 trace.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:OVER:J:ENAB ON* SOUR:DATA:TEL:SDH:OVER:J:PATT B16* SOUR:DATA:TEL:SDH:OVER:J:PATT:B16 "EXFO SONET/SDH"* SOUR:DATA:TEL:SDH:OVER:J:PATT:B16? <p>Returns "EXFO SONET/SDH"</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:ENABled* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:PATtern* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:PATtern:B16

**:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:
J:PATtern:B64**

Description	<p>This command sets the 64 bytes format of the J0 trace.</p> <p>At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern:B64<wsp> <Message></pre>

:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern:B64

Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the 64 bytes sequence of the J0 trace.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:OVER:J:ENAB ON* SOUR:DATA:TEL:SDH:OVER:J:PATT B64* SOUR:DATA:TEL:SDH:OVER:J:PATT:B64 "EXFO SONET/SDH Analyzer Section/RS trace test message"* SOUR:DATA:TEL:SDH:OVER:J:PATT:B64? <p>Returns "EXFO SONET/SDH Analyzer Section/RS trace test message"</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:ENABled* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:PATtern* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:PATtern:B64?

**:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:
J:PATtern:B64?**

Description	This query returns the 64 bytes format of the J0 trace. At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATtern:B64?
Parameter(s)	None
Response Syntax	<Message>

:SOURce[1..n]:DATA:TELEcom:SDH:OVERhead: J:PATTer:n:B64?

Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the 64 bytes format for J0 trace.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:OVER:J:ENAB ON* SOUR:DATA:TEL:SDH:OVER:J:PATT B64* SOUR:DATA:TEL:SDH:OVER:J:PATT:B64 "EXFO SONET/SDH Analyzer Section/RS trace test message"* SOUR:DATA:TEL:SDH:OVER:J:PATT:B64? <p>Returns "EXFO SONET/SDH Analyzer Section/RS trace test message"</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:ENABled* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:PATTern* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:PATTern:B64

**:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:
J:TIM**

Description	<p>This command enables or disables the Trace Identifier Mismatch (TIM) for the expected message defined. Enabled TIM has to give access to the expected trace format and message.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the TIM (Trace Identifier Mismatch).</p>
Example(s)	<pre>* SENS:DATA:TEL:SDH:OVER:J:TIM ON * SENS:DATA:TEL:SDH:OVER:J:TIM? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM?</pre>

**:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:
TIM?**

Description	This query returns the status of the Trace Identifier Mismatch (TIM) for the expected message defined. At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Trace Identifier Mismatch (TIM).
Example(s)	* SENS:DATA:TEL:SDH:OVER:J:TIM ON * SENS:DATA:TEL:SDH:OVER:J:TIM? Returns 1
See Also	* SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM

**:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:
TIM:PATtern**

Description	<p>This command sets the J0 trace in 16 or 64 bytes format.</p> <p>At *RST, this value is set to B16.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATtern<wsp>B16 B64</p>
Parameter(s)	<p>Pattern:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: B16 B64.</p> <p>Sets the J0 value in 16 or 64 bytes format.</p>

**:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:
TIM:PATtern**

B16, selects the 16 bytes format.

B64, selects the 64 bytes format.

Example(s)

* SENS:DATA:TEL:SDH:OVER:J:TIM ON

* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT B16

* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT?

Returns B16

See Also

* SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:
TIM

* SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:
TIM:PATtern?

**:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:
TIM:PATtern?**

Description	This query returns the J0 trace in 16 or 64 bytes format. At *RST, this value is set to B16.
Syntax	:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATtern?
Parameter(s)	None
Response Syntax	<Pattern>

:SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J: TIM:PATtern?

Response(s)	<p>Pattern:</p> <p>The response data syntax for <Pattern> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the J0 value in 16 or 64 bytes format.</p> <p>B16, 16 bytes format is selected.</p> <p>B64, 64 bytes format is selected.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDH:OVER:J:TIM ON* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT B16* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT? <p>Returns B16</p>
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:TIM* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:TIM:PATtern

**:FETCh[1..n]:DATA:TELEcom:SDH:OVERhead:J:
TIM:PATTErn:RECEived?**

Description	This query returns the received J0 trace in 16 or 64 bytes format. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATTErn:RECEived?
Parameter(s)	None
Response Syntax	<Message>

:FETCh[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATTerN:RECEived?

Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the received J0 trace in 16 or 64 bytes format.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:OVER:J:ENAB ON* SOUR:DATA:TEL:SDH:OVER:J:PATT B16* FETC:DATA:TEL:SDH:OVER:J:TIM:PA TT:REC? <p>Returns the J0 value in 16 or 64 bytes format.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:ENABled* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:PA TTern:B16* SOURce[1..n]:DATA:TELEcom:SDH:OVERhead:J:PA TTern:B64

:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATTErn:B16

Description	<p>This command sets the expected message for 16 bytes format of J0 trace.</p> <p>At *RST, this value is set to "EXFO SONET/SDH".</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATTErn:B16<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the expected message for 16 bytes format.</p>
Example(s)	<pre>* SENS:DATA:TEL:SDH:OVER:J:TIM ON * SENS:DATA:TEL:SDH:OVER:J:TIM:PATT B16 * SENS:DATA:TEL:SDH:OVER:J:TIM:PATT:B16 "EXFO SONET/SDH" * SENS:DATA:TEL:SDH:OVER:J:TIM:PATT:B16? Returns "EXFO SONET/SDH"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM * SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATTErn * SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATTErn:B16?</pre>

:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATtern:B16?

Description	This query returns the expected message for 16 bytes format of J0 trace. At *RST, this value is set to "EXFO SONET/SDH".
Syntax	:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATtern:B16?
Parameter(s)	None
Response Syntax	<Message>

**:SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:
TIM:PATTeRn:B16?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the expected message for 16 bytes format.

Example(s)

* SENS:DATA:TEL:SDH:OVER:J:TIM ON

* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT B16

* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT:B16
"EXFO SONET/SDH"

* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT:B16?
Returns "EXFO SONET/SDH"

See Also

* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:
TIM

* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:
TIM:PATTeRn

* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:
TIM:PATTeRn:B16

:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATtern:B64

Description This command sets the expected message for 64 bytes format.

At *RST, the message is "EXFO SONET/SDH Analyzer Section/RS trace test message".

Syntax :SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J:
TIM:PATtern:B64 <wsp> <Message>

**:SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:
TIM:PATTErn:B64**

Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the 64 bytes sequence of the expected J0 trace.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDH:OVER:J:TIM ON* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT B64* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT:B64 <p>"EXFO SONET/SDH Analyzer Section/RS trace test message"</p> <ul style="list-style-type: none">* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT:B64? <p>Returns "EXFO SONET/SDH Analyzer Section/RS trace test message"</p>
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:TIM* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:TIM:PATTErn* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:TIM:PATTErn:B64?

:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATtern:B64?

Description	This query returns the expected message for 64 bytes format. At *RST, this value is set to "EXFO SONET/SDH Analyzer Section/RS trace test message".
Syntax	:SENSe[1..n]:DATA:TELEcom:SDH:OVERhead:J: TIM:PATtern:B64?
Parameter(s)	None
Response Syntax	<Message>

**:SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:
TIM:PATtern:B64?**

Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the expected message for 64 bytes format.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDH:OVER:J:TIM ON* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT B64* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT:B64 <p>"EXFO SONET/SDH Analyzer Section/RS trace test message"</p> <ul style="list-style-type: none">* SENS:DATA:TEL:SDH:OVER:J:TIM:PATT:B64? <p>Returns "EXFO SONET/SDH Analyzer Section/RS trace test message"</p>
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:TIM* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:TIM:PATtern* SENSe[1..n]:DATA:TELecom:SDH:OVERhead:J:TIM:PATtern:B64

**:SOURce[1..n]:DATA:TELEcom:SDH:HOP:PATH:
LABel**

Description This command sets the path signal label (C2) of High Order Path (HOP) for transmitter.

At *RST, the configuration is set to a device-dependent value.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:HOP:PATH:
LABel<wsp>UNEQuipped|EQUipped|TUGS|
LOCKed|AMC3|MEXPeriment|REServed|AM140
|ATMM|MDQDb|FDDim|MHDLc|SSElf|MHLaps
|SSET|M10ETHERENET|GFP|M10FC|AMODuk|
RHPPp|TSIGNal|AISTcm

Parameter(s) Label:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
UNEQuipped|EQUipped|TUGS|LOCKed|AMC3|
MEXPeriment|REServed|AM140|ATMM|MDQDb
|FDDim|MHDLc|SSElf|MHLaps|SSET|
M10ETHERENET|GFP|M10FC|AMODuk|RHPPp|
TSIGNal|AISTcm.
Selects the path signal label.
UNEQuipped, selects the Unequipped path signal label.

**:SOURce[1..n]:DATA:TELEcom:SDH:HOP:PATH:
LABEL**

EQUipped, selects the Equipped non-specific path signal label.

TUGS, selects the TUG (Tributary Unit Group) structure path signal label.

LOCKed, selects the locked TU-n mode path signal label.

AMC3, selects the Asynchronous Mapping for 34M/45M in C-3 path signal label.

MEXPeriment, selects the experimental Mapping path signal label.

REServed, selects the Reserved path signal label.

AM140, selects the Asynchronous Mapping for 140M (DS4NA) path signal label.

ATMM, selects the Mapping for ATM (Asynchronous Transfer Mode) path signal label.

MDQDb, selects the Mapping for DQDB (Distributed Queue Dual Bus) path signal label.

FDDim, selects the Asynchronous Mapping for FDDI (Fiber Distributed Data Interface) path signal label.

MHDLc, selects the mapping of HDLC (High-Level Data Link Control) over SONET path signal label.

SSELf, selects the SSELf with self-synchronization scrambler path signal label.

**:SOURce[1..n]:DATA:TELEcom:SDH:HOP:PATH:
LABel**

MHLaps, selects the Mapping of HDLC (High-Level Data Link Control)/LAPS (Link Access Procedure for SDH) path signal label.

SSET, selects the set SDL with use of a set-reset scrambler path signal label.

M10ETHERNET, selects the 10 Gbps ethernet (IEEE 802.3) path signal label.

GFP, selects the GFP (Generic Framing Procedure) path signal label.

M10FC, selects the Mapping of 10 Gbps FC (Fibre Channel).

AMODuk, selects the Asynchronous Mapping of ODU (Optical Channel Data Unit)k.

RHPPp, selects the Reserved [absolute HDLC (High-Level Data Link Control)/PPP (Point-to-Point Protocol) framed].

TSIGnal, selects the Test signal, ITU-T 0.181 specific mapping path signal label.

AISTcm, selects the VC AIS (Virtual Container - Alarm Indication Signal) (TCM) path signal label.

Example(s)

* SOUR:DATA:TEL:SDH:HOP:PATH:LAB EQU

* SOUR:DATA:TEL:SDH:HOP:PATH:LAB?

Returns EQUIPPED

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:HOP:
PATH: LABel?

**:SOURce[1..n]:DATA:TELEcom:SDH:HOP:PATH:
LABel?**

Description	<p>This query returns the path signal label (C2) of High Order Path (HOP) for transmitter.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABel?
Parameter(s)	None
Response Syntax	<Label>
Response(s)	<p>Label:</p> <p>The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element. Returns the path signal label.</p> <p>UNEQUIPPED, Unequipped path signal label is selected.</p> <p>EQUIPPED, Equipped non-specific path signal label is selected.</p> <p>TUGS, Tributary Unit Group (TUG) structure path signal label is selected.</p> <p>LOCKED, Locked TU-n mode path signal label is selected.</p>

:SOURce[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABEL?

AMC3, Asynchronous Mapping for 34M/45M in C-3 path signal label is selected.

MEXPERIMENT, Experimental mapping path signal label is selected.

RESERVED, Reserved path signal label is selected.

AM140M, Asynchronous Mapping for 140M (DS4NA) path signal label is selected.

ATMM, Mapping for Asynchronous Transfer Mode (ATM) path signal label is selected.

MDQDB, Mapping for Distributed Queue Dual Bus (DQDB) path signal label is selected.

FDDIM, Asynchronous mapping for Fiber Distributed Data Interface (FDDI) path signal label is selected.

MHDLC, Mapping of High-Level Data Link Control (HDLC) over SONET path signal label is selected.

SSELF, SSELF with self-synchronization scrambler path signal label is selected.

MHLAPS, Mapping of High-Level Data Link Control (HDLC)/Link Access Procedure for SDH (LAPS) path signal label is selected.

SSET, Set SDL with use of a set-reset scrambler path signal label is selected.

M10ETHERNET, 10 Gbps ethernet (IEEE 802.3) path signal label is selected.

GFP, Generic Framing Procedure (GFP) path signal label is selected.

M10FC, Mapping of 10 Gbps Fibre Channel (FC) is selected.

**:SOURce[1..n]:DATA:TELEcom:SDH:HOP:PATH:
LABel?**

AMODUK, Asynchronous Mapping of ODUk is selected.

RHPPP, Reserved [absolute High-Level Data Link Control (HDLC)/Point-to-Point Protocol (PPP) framed] is selected.

TSIGNAL, Test signal, ITU-T 0.181 specific mapping path signal label is selected.

AISTCM, Virtual Container - Alarm Indication Signal (VC AIS) (TCM) path signal label is selected.

Example(s)

* SOUR:DATA:TEL:SDH:HOP:PATH:LAB EQU

* SOUR:DATA:TEL:SDH:HOP:PATH:LAB?

Returns EQUIPPED

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:HOP:PATH:
LABel

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

Description This command sets the expected path signal label (C2) of High Order Path (HOP) for receiver.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:SDH:HOP:PATH:
LABel:EXPEcted<wsp>EQUipped|TUGS|
LOCKed|AMC3|MEXPeriment|AM140|ATMM|
MDQDb|FDDim|MHDLc|SSELf|MHLaps|SSET|
M10ETHERNET|GFP|M10FC|AMODuk|RHPPp|
TSIGnal

Parameter(s) Label:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
EQUipped|TUGS|LOCKed|AMC3|MEXPeriment|AM140|ATMM|MDQDb|FDDim|MHDLc|SSELf|MHLaps|SSET|M10ETHERNET|GFP|M10FC|AMODuk|RHPPp|TSIGnal.
Selects the path signal label.
EQUipped, selects the Equipped non-specific path signal label.
TUGS, selects the TUG (Tributary Unit Group) structure path signal label.

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

Parameter(s)	
	LOCKed, selects the Locked TU-n mode path signal label.
	AMC3, selects the Asynchronous Mapping for 34M/45M in C-3 path signal label.
	MEXPeriment, selects the experimental Mapping path signal label.
	AM140, selects the Asynchronous Mapping for 140M (DS4NA) path signal label.
	ATMM, selects the Mapping for ATM (Asynchronous Transfer Mode) path signal label.
	MDQDb, selects the Mapping for DQDB (Distributed Queue Dual Bus) path signal label.
	FDDim, selects the Asynchronous Mapping for FDDI (Fiber Distributed Data Interface) path signal label.
	MHDLC, selects the mapping of HDLC (High-Level Data Link Control) over SONET path signal label.
	SSELf, selects the SSELf with self-synchronization scrambler path signal label.
	MHLaps, selects the Mapping of HDLC (High-Level Data Link Control)/LAPS (Link Access Procedure for SDH) path signal label.
	SSET, selects the set SDL with use of a set-reset scrambler path signal label.
	M10ETHERNET, selects the 10 Gbps ethernet (IEEE 802.3) path signal label.
	GFP, selects the GFP (Generic Framing Procedure) path signal label.
	M10FC, selects the Mapping of 10 Gbps FC (Fibre Channel).

**:SENSe[1..n]:DATA:TELecom:SDH:HOP:PATH:
LABel:EXPeCted**

AMODuk, selects the Asynchronous Mapping of ODUk.

RHPPp, selects the Reserved [absolute HDLC (High-Level Data Link Control)/PPP (Point-to-Point Protocol) framed].

TSIGnal, selects the Test signal, ITU-T 0.181 specific mapping path signal label.

Example(s)

* SENS:DATA:TEL:SDH:HOP:PATH:LAB:EXP EQU
* SENS:DATA:TEL:SDH:HOP:PATH:LAB:EXP?
Returns EQUIPPED

See Also

* SENSe[1..n]:DATA:TELecom:SDH:HOP:PATH:
LABel:EXPeCted?

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

Description	<p>This query returns the expected path signal label (C2) of High Order Path (HOP) for receiver.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABel:EXPEcted?
Parameter(s)	None
Response Syntax	<Label>
Response(s)	<p>Label:</p> <p>The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the path signal label.</p> <p>EQUIPPED, Equipped non-specific path signal label is selected.</p> <p>TUGS, Tributary Unit Group (TUG) structure path signal label is selected.</p> <p>LOCKED, Locked TU-n mode path signal label is selected.</p> <p>AMC3, Asynchronous Mapping for 34M/45M in C-3 path signal label is selected.</p>

:SENSE[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABel:EXPEcted?

MEXPERIMENT, Experimental mapping path signal label is selected.

AM140, Asynchronous Mapping for 140M (DS4NA) path signal label is selected.

ATMM, Mapping for Asynchronous Transfer Mode (ATM) path signal label is selected.

MDQDB, Mapping for Distributed Queue Dual Bus (DQDB) path signal label is selected.

FDDIM, Asynchronous Mapping for Fiber Distributed Data Interface (FDDI) path signal label is selected.

MHDLC, Mapping of High-Level Data Link Control (HDLC) over SONET path signal label is selected.

SSELF, SSELF with self-synchronization scrambler path signal label is selected.

MHLAPS, Mapping of High-Level Data Link Control (HDLC)/Link Access Procedure for SDH (LAPS) path signal label is selected.

SSET, Set SDL with use of a set-reset scrambler path signal label is selected.

M10ETHERNET, 10 Gbps ethernet (IEEE 802.3) path signal label is selected.

GFP, Generic Framing Procedure (GFP) path signal label is selected.

M10FC, Mapping of 10 Gbps Fibre Channel (FC) is selected.

AMODUK, Asynchronous Mapping of ODUk is selected.

RHPPP, Reserved [absolute High-Level Data Link Control (HDLC)/Point-to-Point Protocol (PPP) framed] is selected.

**:SENSe[1..n]:DATA:TELecom:SDH:HOP:PATH:
LABel:EXPEcted?**

TSIGNAL, Test signal, ITU-T 0.181 specific mapping path signal label is selected.

Example(s)

* SENS:DATA:TEL:SDH:HOP:PATH:LAB:EXP EQU
* SENS:DATA:TEL:SDH:HOP:PATH:LAB:EXP?
Returns EQUIPPED

See Also

* SENSe[1..n]:DATA:TELecom:SDH:HOP:PATH:
LABel:EXPEcted

**:FETCh[1..n]:DATA:TELEcom:SDH:HOP:PATH:
LABel?**

Description	<p>This query returns the path signal label of High Order Path (HOP) for receiver.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABel?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Label></p>
Response(s)	<p>Label: The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element. Returns the path signal label.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:HOP:PATH:LAB EQU * FETC:DATA:TEL:SDH:HOP:PATH:LAB? Returns the path signal label.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:HOP:PATH: LABel</p>

:SENSe[1..n]:DATA:TELEcom:SDH:HOP:PUNeq

Description	<p>This command enables or disables the Payload Label Mismatch / Unequipped path for the expected message defined of High Order Path (HOP).</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDH:HOP:PUNeq <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Payload Label Mismatch / Unequipped path of HOP (High Order Path).</p>
Example(s)	<pre>* SENS:DATA:TEL:SDH:HOP:PUN ON * SENS:DATA:TEL:SDH:HOP:PUN? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:SDH:HOP:PUNeq?</pre>

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

Description	This query returns the status of Payload Label Mismatch / Unequipped for the expected message defined of High Order Path (HOP). At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:SDH:HOP:PUNeq?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Payload Label Mismatch / Unequipped path of High Order Path (HOP).
Example(s)	* SENS:DATA:TEL:SDH:HOP:PUN ON * SENS:DATA:TEL:SDH:HOP:PUN? Returns 1
See Also	* SENSe[1..n]:DATA:TELEcom:SDH:HOP:PUNeq

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:TYPE

Description	<p>This command selects the type of High Order Path (HOP) alarm.</p> <p>At *RST, this value is set to AUAis.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:TYPE<wsp>AUAis HPRDi ESD ECD EPD H4LOM AULop HPUNeq</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AUAis HPRDi ESD ECD EPD H4LOM AULop HPUNeq.</p> <p>Selects the type of High Order Path (HOP) alarm.</p> <p>AUAis, selects the AU-AIS (Administrative Unit - Alarm Indication Signal) which generates an all-ones patterns for the path and payload.</p> <p>HPRDi, selects the HP-RDI (High Order Path - Remote Defect Indication) which generates a "100" pattern for bits 5, 6 and 7 of the G1 byte.</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARM: HOP:PATH:TYPE

ESD, selects the ERDI-SD (Enhanced RDI - Server Defect) which generates a "101" pattern for the bits 5, 6 and 7 of the G1 byte.

ECD, selects the ERDI-CD (Enhanced RDI - Connectivity Defect) which generates a "110" pattern for the bits 5, 6 and 7 of the G1 byte.

EPD, selects the ERDI-PD (Enhanced RDI - Payload Defect) which generates a "010" pattern for the bits 5, 6 and 7 of the G1 byte.

H4LOM, selects the H4-LOM (H4 - Loss of Multiframe) (available with TU-11, TU-12 and TU-2) which generates a wrong H4 byte multiframe indicator sequence.

AULop, selects the AU-LOP (Administrative Unit - Loss of Pointer) which generates a non-valid pointer.

HPUNeq, selects the HP-UNEQ (High Order Path - Unequipped) which generates unequipped signal labels (path and payload are set to "00 H").

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:TYPE**

Example(s) * SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:TYPE
AUA
* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:TYPE?
Returns AUAIS

Note For **8120NGE/8130NGE/8130NGEv2** modules,
choices are AUAI|HPRDi|ESD|ECD|EPD|
H4LOM|AULop|HPUNeq.
For **8140** module, choices are AUAI|HPRDi|
ESD|ECD|EPD|AULop|HPUNeq.

See Also * SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:TYPE?**

Description	<p>This query returns the type of High Order Path (HOP) alarm.</p> <p>At *RST, this value is set to AUAis.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:TYPE?</code>
Parameter(s)	None
Response Syntax	<code><Alarm></code>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <code><Alarm></code> is defined as a <code><CHARACTER RESPONSE DATA></code> element. Returns the type of High Order Path (HOP) alarm. AUAIS, Administrative Unit - Alarm Indication Signal (AU-AIS) is selected as High Order Path (HOP) alarm.</p> <p>HPRDI, High Order Path - Remote Defect Indication (HPRDI) is selected as High Order Path (HOP) alarm.</p> <p>ESD, Enhanced RDI - Server Defect (ERDI-SD) is selected as High Order Path (HOP) alarm.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:TYPE?**

ECD, Enhanced RDI - Connectivity Defect (ERDI-CD) is selected as High Order Path (HOP) alarm.

EPD, Enhanced RDI - Payload Defect (ERDI-PD) is selected as High Order Path (HOP) alarm.

H4LOM, H4 - Loss of Multiframe (H4-LOM) is selected as High Order Path (HOP) alarm.

AULOP, Administrative Unit - Loss of Pointer (AU-LOP) is selected as High Order Path (HOP) alarm.

HPUNEQ, High Order Path - Unequipped (HP-UNEQ) is selected as High Order Path (HOP) alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:TYPE
AUA

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:TYPE?
Returns AUAIS

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:TYPE

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH**

Description	<p>This command enables or disables the High Order Path (HOP) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH<wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables and disables the HOP (High Order Path) alarm generation.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:TYPE AUA * SOUR:DATA:TEL:SDH:ALAR:HOP:PATH ON * SOUR:DATA:TEL:SDH:ALAR:HOP:PATH? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH?**

Description	This query returns the status of High Order Path (HOP) alarm generation. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH?**

Response(s)

Set:

The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of High Order Path (HOP) alarm generation.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:TYPE
AUA

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH ON

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:MANual:TYPE**

Description	<p>This command sets the manual type of High Order Path (HOP) error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:MANual:TYPE<wsp>BERRor HPRei</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor HPRei.</p> <p>Selects the mode for HOP (High Order Path) error generation.</p> <p>BERRor, selects the type of High Order Path (HOP) error as B3.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:MANual:TYPE**

HPRei, selects the type of High Order Path (HOP) error as HP-REI (High Order path - Remote Error Indicator).

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE BERR

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE? Returns BERROR

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:MANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:MANual:TYPE?**

Description	This query returns the manual type of High Order Path (HOP) error. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:MANual:TYPE?

Response(s)

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of High Order Path (HOP) error. BERROR, B3 is selected as High Order Path (HOP) error. HPREI, High Order path - Remote Error Indicator (HP-REI) is selected as High Order Path (HOP) error.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE BERR

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE? Returns BERROR

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AMOut**

Description	<p>This command sets the amount of High Order Path (HOP) error to be injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AMOut <wsp> <Amount> MAXimum MINimum</pre>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AMOUNT**

Sets the amount of HOP (High Order Path) error to be injected.

Choices are 1 through 50.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE BERR

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AMO 15

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AMO?

Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:MANual:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AMOUNT?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AMOUnt?**

Description	<p>This query returns the amount of High Order Path (HOP) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AMOUnt?[<wsp>MAXimum MINimum]</code>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<code><Amount></code>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AMOut?

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of High Order Path (HOP) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AMO 15 * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AMO? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AMOut

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:INJect**

Description	<p>This command injects the type of High Order Path (HOP) error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN: TYPE BERR * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AMO 15 * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:INJ
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:AMOUNT

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:TYPE**

Description	<p>This command selects the type of High Order Path (HOP) error for automated injection.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AUTomated:TYPE<wsp>BERRor HPRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor HPRei. Selects the type of High Order Path (HOP) error for automated injection.</p> <p>BERRor, selects the type of High Order Path (HOP) error as B3.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:TYPE**

HPRei, selects the type of High Order Path (HOP) error as HP-REI (High Order path - Remote Error Indicator).

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
TYPE BERR
* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
TYPE? Returns BERROR

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated:TYPE?
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AUTomated:TYPE?

Description	This query returns the type of High Order Path (HOP) error for automated injection. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:TYPE?**

Response(s)

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the type of High Order Path (HOP) error for the automated injection.

BERROR, B3 is selected as High Order Path (HOP) error.

HPREI, High Order path - Remote Error Indicator (HP-REI) is selected as High Order Path (HOP) error.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:TYPE BERR

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:TYPE? Returns BERROR

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AUTomated:RATE

Description	<p>This command sets the injection rate for the selected High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AUTomated:RATE<wsp> <Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:RATE**

Sets the injection rate for the selected High Order Path (HOP) error.

Example(s)

```
* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
TYPE BERR
* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
RATE 1.0E-10
* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
RATE? Returns 1.0E-10
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated:RATE?
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated
```

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AUTomated:RATE?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injection rate will be returned.</p>
Response Syntax	<p><Rate></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected High Order Path (HOP) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:RATE 1.0E-10</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:RATE? Returns 1.0E-10</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated**

Description

This command enables or disables the selected automated High Order Path (HOP) error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated <wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated High Order Path (HOP) error injection.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:RATE 1.0E-10</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT ON</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT?</p> <p>Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:AUTomated?</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AUTomated?

Description	This query returns the status of automated High Order Path (HOP) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of automated High Order Path (HOP) error injection.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
TYPE BERR

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
RATE 1.0E-10

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT ON

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:AUTomated:CONTInuous

Description

This command enables or disables the continuous rate of automated High Order Path (HOP) error injection.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:CONTInuous <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:CONTInuous****Parameter(s)**

Set:

The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated High Order Path (HOP) error injection continuously.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
TYPE BERR
* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
CONT ON
* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
CONT? Returns 1
* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT ON

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:CONTInuous?**

Description This query returns the status of continuous rate of automated High Order Path (HOP) error injection.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:CONTInuous?

Parameter(s) None

Response Syntax <Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:AUTomated:CONTInuous?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of continuous rate of automated High Order Path (HOP) error injection.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
TYPE BERR

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
CONT ON

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT:
CONT? Returns 1

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AUT ON

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:AUTomated:CONTInuous

:SOURce[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel

Description	<p>This command sets the path signal label (V5) of Low Order Path (LOP) for transmitter.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel<wsp>UNEQipped EQUipped ASYNchronous BISYNch BYSYNch SIGNal TEST VCAis</pre>
Parameter(s)	<p>Label:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>UNEQipped EQUipped ASYNchronous BISYNch BYSYNch SIGNal TEST VCAis.</pre> <p>Selects the path signal label.</p> <p>UNEQipped, selects the Unequipped path signal label.</p> <p>EQUipped, selects the Equipped non-specific path signal label.</p> <p>ASYNchronous, selects the Asynchronous path signal label.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:LOP:PATH:
LABel**

BISynch, selects the Bit Synchronous path signal label.

BYSynch, selects the Byte Synchronous path signal label.

SIGNal, selects the extended path Signal.

TEST, selects the Test signal, ITU-T 0.181 specific mapping path signal label.

VCAis, selects the VC AIS (Virtual Container - Alarm Indication Signal) (TCM).

Example(s)

* SOUR:DATA:TEL:SDH:LOP:PATH:LAB EQU

* SOUR:DATA:TEL:SDH:LOP:PATH:LAB?

Returns EQUIPPED

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:LOP:PATH:
LABel?

**:SOURce[1..n]:DATA:TELEcom:SDH:LOP:PATH:
LABel?**

Description	<p>This query returns the path signal label (V5) of Low Order Path (LOP) for transmitter.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Label></p>
Response(s)	<p>Label: The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element. Returns the path signal label is selected. UNEQUIPPED, Unequipped path signal label is selected. EQUIPPED, Equipped non-specific path signal label is selected. ASYNCHRONOUS, Asynchronous path signal label is selected. BISYNCH, Bit Synchronous path signal label is selected.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:LOP:PATH:
LABel?**

BYSYNCH, Byte Synchronous path signal label is selected.

SIGNAL, Extended path Signal label is selected.

TEST, Test signal, ITU-T 0.181 specific mapping path signal label is selected.

VCAIS, Virtual Container - Alarm Indication Signal (VC AIS)(TCM) is selected.

Example(s)

* SOUR:DATA:TEL:SDH:LOP:PATH:LAB EQU

* SOUR:DATA:TEL:SDH:LOP:PATH:LAB?

Returns EQUIPPED

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:LOP:PATH:
LABel

:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel:EXPeCted

Description	<p>This command sets the expected path signal label (V5) of Low Order Path (LOP) for the receiver.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel:EXPeCted<wsp>EQUipped ASYNchronous BISYNch BYSYnch SIGNal TEST</pre>
Parameter(s)	<p>Label:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>EQUipped ASYNchronous BISYNch BYSYnch SIGNal TEST.</pre> <p>Selects the path signal label.</p> <p>EQUipped, selects the Equipped non-specific path signal label.</p> <p>ASYNchronous, selects the Asynchronous path signal label.</p> <p>BISYNch, selects the Bit Synchronous path signal label.</p>

**:SENSe[1..n]:DATA:TELecom:SDH:LOP:PATH:
LABel:EXPEcted**

BYSYnch, selects the Byte Synchronous path signal label.

SIGNal, selects the extended path Signal.

TEST, selects the Test signal, ITU-T 0.181 specific mapping path signal label.

Example(s)

* SENS:DATA:TEL:SDH:LOP:PATH:LAB:EXP EQU
* SENS:DATA:TEL:SDH:LOP:PATH:LAB:EXP?
Returns EQUIPPED

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SENSe[1..n]:DATA:TELecom:SDH:LOP:PATH:
LABel:EXPEcted?

**:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PATH:
LABel:EXPEcted?**

Description	<p>This query returns the expected path signal label (V5) of Low Order Path (LOP) for the receiver.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel:EXPEcted?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Label></p>
Response(s)	<p>Label: The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element. Returns the path signal label. EQUIPPED, Equipped non-specific path signal label is selected. ASYNCHRONOUS, Asynchronous path signal label is selected. BISYNCH, Bit Synchronous path signal label is selected. BYSYNCH, Byte Synchronous path signal label is selected.</p>

**:SENSe[1..n]:DATA:TELecom:SDH:LOP:PATH:
LABel:EXPEcted?**

SIGNAL, Extended path signal is selected.
TEST, ITU-T 0.181 specific mapping path signal
label is selected.

Example(s)

* SENS:DATA:TEL:SDH:LOP:PATH:LAB:EXP EQU
* SENS:DATA:TEL:SDH:LOP:PATH:LAB:EXP?
Returns EQUIPPED

Note

FTB/IQS-8140 Transport Blazer does not support
this query.

See Also

* SENSe[1..n]:DATA:TELecom:SDH:LOP:PATH:
LABel:EXPEcted

**:FETCh[1..n]:DATA:TELEcom:SDH:LOP:PATH:
LABel?**

Description	<p>This query returns the path signal label of Low Order Path (LOP) for the receiver.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDH:LOP:PATH: LABel?
Parameter(s)	None
Response Syntax	<Label>
Response(s)	<p>Label:</p> <p>The response data syntax for <Label> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the path signal label for the receiver.</p>

**:FETCh[1..n]:DATA:TELEcom:SDH:LOP:PATH:
LABel?**

Example(s)	* SOUR:DATA:TEL:SDH:LOP:PATH:LAB EQU * FETC:DATA:TEL:SDH:LOP:PATH:LAB? Returns the path signal label.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:LOP:PATH :LABel

:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PUNeq

Description	<p>This command enables or disables the Payload Label Mismatch / Unequipped path for the expected message of Low Order Path (LOP).</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PUNeq <wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Payload Label Mismatch / Unequipped path for the expected message of LOP (Low Order Path).</p>
Example(s)	<p>* SENS:DATA:TEL:SDH:LOP:PUN ON * SENS:DATA:TEL:SDH:LOP:PUN? Returns 1</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this command.</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:SDH:LOP:PUNeq?</p>

:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PUNeq?

Description	This query returns the status of Payload Label Mismatch / Unequipped path for the expected message of Low Order Path (LOP). At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:SDH:LOP:PUNeq?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Payload Label Mismatch / Unequipped path of Low Order Path (LOP).
Example(s)	* SENS:DATA:TEL:SDH:LOP:PUN ON * SENS:DATA:TEL:SDH:LOP:PUN? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SENSe[1..n]:DATA:TELEcom:SDH:LOP:PUNeq

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:TYPE**

Description	<p>This command selects the type of Low Order Path (LOP) alarm.</p> <p>At *RST, this value is set to TUAis.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:TYPE<wsp>TUAis LPRDi ESD ECD EPD LPRFi TULop LPUNeq</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TUAis LPRDi ESD ECD EPD LPRFi TULop LPUNeq.</p> <p>Selects the type of LOP (Low Order Path) alarm.</p> <p>TUAis, selects the TU-AIS (Tributary Unit - Alarm Indication Signal) which generates an all-ones pattern for the V1 and V2 bytes of the TU path and payload.</p> <p>LPRDi, selects the LP-RDI (Low Order Path - Remote Defect Indication) which generates "1" for the bit 8 of the V5 byte and a "00" pattern for bits 6 and 7 of the K4 byte.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:TYPE**

ESD, selects the ERDI-SD (Enhanced RDI - Server Defect) which generates a "101" pattern for bits 5, 6, and 7 of the K4 byte, and "1" for bit 8 of the V5 byte.

ECD, selects the ERDI-CD (Enhanced RDI - Connectivity Defect) which generates a "110" pattern for bits 5, 6, and 7 of the K4 byte, and "1" for bit 8 of the V5 byte.

EPD, selects the ERDI-PD (Enhanced RDI - Path Payload Defect) which generates a "010" pattern for bits 5, 6, and 7 of the K4 byte, and "0" for bit 8 of the V5 byte.

LPRFi, selects the LP-RFI (Low Order Path - Remote Failure Indication) (available with VC-11 only) which generates "1" for the bit 4 of the V5 byte.

TULop, selects the TU-LOP (Tributary Unit - Loss of Pointer) which generates a non-valid pointer.

LPUNeq, selects the LP-UNEQ (Low Order Path - Unequipped) which generates unequipped LP signal label (bits 5 through 7 of V5 byte are set to "000").

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:TYPE**

Example(s)	* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE TUA * SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE? Returns TUAIS
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:TYPE?**

Description	<p>This query returns the type of Low Order Path (LOP) alarm.</p> <p>At *RST, this value is set to TUAIs.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Low Order Path (LOP) alarm. TUAIs, Tributary Unit - Alarm Indication Signal (TU-AIS) is selected as Low Order Path (LOP) alarm.</p> <p>LPRDI, Tributary Unit - Alarm Indication Signal (LP-RDI) is selected as Low Order Path (LOP) alarm.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:TYPE?**

ESD, Enhanced RDI - Server Defect (ERDI-SD) is selected as Low Order Path (LOP) alarm.

ECD, Enhanced RDI - Connectivity Defect (ERDI-CD) is selected as Low Order Path (LOP) alarm.

EPD, Enhanced RDI - Path Payload Defect (ERDI-PD) is selected as Low Order Path (LOP) alarm.

LPRFI, Low Order Path - Remote Failure Indication (LP-RFI) is selected as Low Order Path (LOP) alarm.

TULOP, Tributary Unit - Loss of Pointer (TU-LOP) is selected as Low Order Path (LOP) alarm.

LPUNEQ, Low Order Path - Unequipped (LP-UNEQ) is selected as Low Order Path (LOP) alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE
TUA

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE?
Returns TUAIS

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:TYPE

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH**

Description	<p>This command enables or disables the Low Order Path (LOP) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH<wsp><Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables and disables the LOP (Low Order Path) alarm generation.</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE TUA* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH ON* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH?**

Description	<p>This query returns the status of Low Order Path (LOP) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Low Order Path (LOP) alarm generation.</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE TUA* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH ON* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:MANual:TYPE**

Description	<p>This command selects the manual type of Low Order Path (LOP) error.</p> <p>At *RST, this value is set to BIP2.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:MANual:TYPE<wsp>BIP2 LPRei</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 LPRei.</p> <p>Selects the type of Low Order Path (LOP) error. BIP2, selects the Low Order Path (LOP) error as BIP-2 (Bit-Interleave Parity - 2 bits).</p> <p>LPRei, selects the Low Order Path (LOP) error as LP-REI (Low Order Path - Remote Error Indicator).</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:MANual:TYPE**

Example(s) * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN:
TYPE BIP2
* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN:
TYPE? Returns BIP2

Note FTB/IQS-8140 Transport Blazer does not support
this command.

See Also * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:MANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:MANual:TYPE?**

Description	<p>This query returns the manual type of Low Order Path (LOP) error.</p> <p>At *RST, this value is set to BIP2.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Low Order Path (LOP) error. BIP2, Bit-Interleave Parity - 2 bits (BIP2) is selected as Low Order Path (LOP) error. LPREI, Low Order Path - Remote Error Indicator (LP-REI) is selected as Low Order Path (LOP) error.</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:MANual:TYPE?

Example(s)	* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN: TYPE BIP2 * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN: TYPE? Returns BIP2
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AMOut**

Description	<p>This command sets the amount of Low Order Path (LOP) error to be injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AMOut <wsp> <Amount> MAXimum MINimum</pre>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the amount of LOP (Low Order Path) error. Choices are 1 through 50.</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AMOut

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN:TYPE BIP2* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AMO 15* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AMO? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AMOut?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AMOut?<wsp>MAXimum|MINimum]**

Description	<p>This query returns the amount of Low Order Path (LOP) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AMOut?<wsp>MAXimum MINimum]</code>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<code><Amount></code>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AMOut?

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Low Order Path (LOP) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN:TYPE BIP2</p> <p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AMO 15</p> <p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AMO?</p> <p>Returns 15</p>
Note	<p>FTB/QS-8140 Transport Blazer does not support this query.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AMOut</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:INJect**

Description	<p>This command injects the type of Low Order Path (LOP) error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN: TYPE BIP2 * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AMO 15 * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:INJ
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AMOut

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated:TYPE

Description	<p>This command selects the type of Low Order Path (LOP) error for automated injection.</p> <p>At *RST, this value is set to BIP2.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated:TYPE<wsp>BIP2 LPRei</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 LPRei.</p> <p>Selects the type of Low Order Path (LOP) error for automated injection.</p> <p>BIP2, selects the Low Order Path (LOP) error as BIP-2 (Bit-Interleave Parity - 2 bits).</p>
Parameter(s)	<p>LPRei, selects the Low Order Path (LOP) error as LOP-REI (Low Order Path - Remote Error Indicator).</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated:TYPE**

Example(s)	* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT: TYPE BIP2 * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT: TYPE? Returns BIP2
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AUTomated:TYPE? * SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AUTomated

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated:TYPE?

Description	<p>This query returns the type of Low Order Path (LOP) error for automated injection.</p> <p>At *RST, this value is set to BIP2.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of Low Order Path (LOP) error for the automated injection.</p> <p>BIP2, Bit-Interleave Parity - 2 bits (BIP2) is selected as Low Order Path (LOP) error.</p> <p>LPREI, Low Order Path - Remote Error Indicator (LP-REI) is selected as Low Order Path (LOP) error.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated:TYPE?****Example(s)**

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:
TYPE BIP2
* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:
TYPE? Returns BIP2

Note

FTB/IQS-8140 Transport Blazer does not support
this query.

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated:RATE**

Description	<p>This command sets the injection rate for the selected Low Order Path (LOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated:RATE<wsp> <Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected Low Order Path (LOP) error.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated:RATE****Example(s)**

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:
TYPE BIP2
* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:
RATE 1.0E-10
* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:
RATE? Returns 1.0E-10

Note

FTB/IQS-8140 Transport Blazer does not support
this command.

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:AUTomated:RATE?
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Low Order Path (LOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated:RATE?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injection rate will be returned.</p>
Response Syntax	<p><Rate></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Low Order Path (LOP) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:TYPE BIP2* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:RATE 1.0E-10* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:RATE? Returns 1.0E-10
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated

Description	<p>This command enables or disables the selected automated Low Order Path (LOP) error at the rate specified or continuously.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated Low Order Path (LOP) error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT: TYPE BIP2* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT: RATE 1.0E-10* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT ON* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated?**

Description	<p>This query returns the status of automated Low Order Path (LOP) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated Low Order Path (LOP) error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT: TYPE BIP2* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT: RATE 1.0E-10* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT ON* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AUTomated

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated:CONTInuous

Description	<p>This command enables or disables the continuous rate of automated Low Order Path (LOP) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated:CONTInuous <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated Low Order Path (LOP) error injection continuously.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated:CONTInuous**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:TYPE BIP2* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:CONT ON* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:CONT? Returns 1* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT ON
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AUTomated:CONTInuous?

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated:CONTInuous?

Description	<p>This query returns the status of continuous rate of automated Low Order Path (LOP) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:AUTomated:CONTInuous?</code>
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of continuous rate of automated Low Order Path (LOP) error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:AUTomated:CONTInuous?****Example(s)**

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:
TYPE BIP2
* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:
CONT ON
* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT:
CONT? Returns 1
* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AUT ON

Note

FTB/IQS-8140 Transport Blazer does not support
this query.

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:AUTomated
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:AUTomated:CONTInuous

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:HISTory?**

Description	<p>This query returns the history status of High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:HISTory?<wsp>BERRor HPRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor HPRei. Selects the type of HOP (High Order Path) error. BERRor, selects the type of High Order Path (HOP) error as B3. HPRei, selects the type of High Order Path (HOP) error as HP-REI (High Order path - Remote Error Indicator).</p>
Response Syntax	<p><History></p>

**:FETCh[1..n]:DATA:TELecom:SDH:ERRor:HOP:
PATH:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AMO 15* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:INJ* FETC:DATA:TEL:SDH:ERR:HOP:PATH:HIST? BERR Returns the error history status.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:MANual:TYPE* SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:AMOUNT* SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:INJECT

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:SEConds?**

Description This query returns the number of seconds within which High Order Path (HOP) error occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:SEConds? <wsp>BERRor|HPRei

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor|HPRei.
Selects the type of HOP (High Order Path) error.
BERRor, selects the type of High Order Path (HOP) error as B3.
HPRei, selects the type of High Order Path (HOP) error as HP-REI (High Order path - Remote Error Indicator).

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELecom:SDH:ERRor:HOP:
PATH:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of High Order Path (HOP) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AMO 15 * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:INJ * FETC:DATA:TEL:SDH:ERR:HOP:PATH:SEC? BERR Returns the number of errored seconds.
See Also	* SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:AMOUNT * SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:CURRent?**

Description	<p>This query returns the current status for High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:CURRent?<wsp>BERRor HPRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor HPRei. Selects the type of HOP (High Order Path) error. BERRor, selects the type of High Order Path (HOP) error as B3.</p> <p>HPRei, selects the type of High Order Path (HOP) error as HP-REI (High Order path - Remote Error Indicator).</p>
Response Syntax	<p><Current></p>

:FETCh[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:CURRent?

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none"> * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AMO 15 * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:INJ * FETC:DATA:TEL:SDH:ERR:HOP:PATH:CURR? BERR Returns the current error status.
See Also	<ul style="list-style-type: none"> * SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:AMOUNT * SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:INJECT

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:COUNT?**

Description	<p>This query returns the count of High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:COUNT? <wsp>BERRor HPRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor HPRei. Selects the type of HOP (High Order Path) error. BERRor, selects the type of High Order Path (HOP) error as B3. HPRei, selects the type of High Order Path (HOP) error as HP-REI (High Order path - Remote Error Indicator).</p>
Response Syntax	<p><Count></p>

**:FETCh[1..n]:DATA:TELecom:SDH:ERRor:HOP:
PATH:COUNT?**

Response(s)	Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of High Order Path (HOP) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AMO 15 * SOUR:DATA:TEL:SDH:ERR:HOP:PATH:INJ * FETC:DATA:TEL:SDH:ERR:HOP:PATH:COUN? BERR Returns current error count.
See Also	* SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:MANual:TYPE * SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:AMOUNT * SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:RATE?**

Description	<p>This query returns the rate of High Order Path (HOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:RATE?<wsp>BERRor HPRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor HPRei. Selects the type of HOP (High Order Path) error. BERRor, selects the type of High Order Path (HOP) error as B3. HPRei, selects the type of High Order Path (HOP) error as HP-REI (High Order path - Remote Error Indicator).</p>
Response Syntax	<p><Rate></p>

**:FETCh[1..n]:DATA:TELecom:SDH:ERRor:HOP:
PATH:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the rate of High Order Path (HOP) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:MAN:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:AMO 15* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:INJ* FETC:DATA:TEL:SDH:ERR:HOP:PATH:RATE? BERR Returns current error rate.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:MANual:TYPE* SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:AMOUNT* SOURce[1..n]:DATA:TELecom:SDH:ERRor:HOP:PATH:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:HISTory?**

Description

This query returns the history status of High Order Path (HOP) alarm.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:HISTory? <wsp>AUAis|HPRDi|ESD|ECD|E
PD|H4LOM|AULop|HPUNeq|HPTim|HPPLm

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

AUAis|HPRDi|ESD|ECD|EPD|H4LOM|AULop|
HPUNeq|HPTim|HPPLm.

Selects the type of HOP (High Order Path) alarm.

AUAis, selects the AU-AIS (Administrative Unit - Alarm Indication Signal) when the H1 and H2 bytes contain an all-ones pattern in three consecutive frames.

HPRDi, selects the HP-RDI (High Order Path - Remote Defect Indication) alarm, when bits 5, 6 and 7 of the G1 byte contain the "100" or "111" pattern in five consecutive frames.

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:HISTory?**

ESD, selects the ERDI-SD (Enhanced RDI - Server Defect) when bits 5, 6 and 7 of the G1 byte contain the "101" pattern in five consecutive frames.

ECD, selects the ERDI-CD (Enhanced RDI - Connectivity Defect) alarm, when bits 5, 6 and 7 of the G1 byte contain the "110" pattern in five consecutive frames.

EPD, selects the ERDI-PD (Enhanced RDI - Payload Defect) alarm, when bits 5, 6 and 7 of the G1 byte contain the "010" pattern in five consecutive frames.

H4LOM, selects the H4-LOM (H4 - Loss of Multiframe), for TU structured optical frames, the H4-LOM alarm indicates that the system loss track of the H4 byte multiframe indicator sequence.

AULop, selects the AU-LOP (Administrative Unit - Loss of Pointer), for non-concatenated payloads, the AU-LOP alarm indicates that a valid pointer is not found in N consecutive frames (where $8 < N <= 10$), or N consecutive NDFs ("1001" pattern) are detected.

HPUNeq, selects the HP-UNEQ (High Order Path - Unequipped) when the C2 bytes contain "00 H" in five consecutive frames.

HPTim, selects the HP-TIM (High Order Path - Trace Identifier Mismatch) which indicates that none of the sampled path trace strings match the expected message value.

**:FETCh[1..n]:DATA:TELeom:SDH:ALARm:HOP:
PATH:HISTory?**

The HP-TIM alarm result is only available when HP-TIM from J1 Trace section has been enabled. HPPLm, selects the HP-PLM (High Order Path - Payload Label Mismatch), declared upon receipt of five consecutive frames with mismatched VC (Virtual Container) signal labels.

Response Syntax <History>

Response(s) History:
The response data syntax for <History> 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.

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:HISTory?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:TYPE AUA* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH ON* FETC:DATA:TEL:SDH:ALAR:HOP:PATH:HIST? AUA Returns the alarm history status.
Note	For 8120NGE/8130NGE/8130NGEv2 modules, choices are AUAis HPRDi ESD ECD EPD H4LOM AULop HPUNeq HPTim HPPLm. For 8140 module, choices are AUAis HPRDi ESD ECD EPD AULop HPUNeq HPTim HPPLm.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:HOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:HOP:PATH

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:SEConds?**

Description This query returns the number of seconds within which High Order Path (HOP) alarm occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:SEConds? <wsp>AUAis|HPRDi|ESD|ECD|
EPD|H4LOM|AULop|HPUNeq|HPTim|HPPLm

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
AUAis|HPRDi|ESD|ECD|EPD|H4LOM|AULop|HPUNeq|HPTim|HPPLm.
Selects the type of HOP (High Order Path) alarm.
AUAis, selects the AU-AIS (Administrative Unit - Alarm Indication Signal) when the H1 and H2 bytes contain an all-ones pattern in three consecutive frames.
HPRDi, selects the HP-RDI (High Order Path - Remote Defect Indication) alarm, when bits 5, 6 and 7 of the G1 byte contain the "100" or "111" pattern in five consecutive frames.

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:SEConds?**

ESD, selects the ERDI-SD (Enhanced RDI - Server Defect) when bits 5, 6 and 7 of the G1 byte contain the "101" pattern in five consecutive frames.

ECD, selects the ERDI-CD (Enhanced RDI - Connectivity Defect) alarm, when bits 5, 6 and 7 of the G1 byte contain the "110" pattern in five consecutive frames.

EPD, selects the ERDI-PD (Enhanced RDI - Payload Defect) alarm, when bits 5, 6 and 7 of the G1 byte contain the "010" pattern in five consecutive frames.

H4LOM, selects the H4-LOM (H4 - Loss of Multiframe), for TU structured optical frames, the H4-LOM alarm indicates that the system loss track of the H4 byte multiframe indicator sequence.

AULop, selects the AU-LOP (Administrative Unit - Loss of Pointer), for non-concatenated payloads, the AU-LOP alarm indicates that a valid pointer is not found in N consecutive frames (where $8 < N <= 10$), or N consecutive NDFs ("1001" pattern) are detected.

HPUNeq, selects the HP-UNEQ (High Order Path - Unequipped) when the C2 bytes contain "00 H" in five consecutive frames.

HPTim, selects the HP-TIM (High Order Path - Trace Identifier Mismatch) which indicates that none of the sampled path trace strings match the expected message value.

**:FETCh[1..n]:DATA:TELecom:SDH:ALARm:HOP:
PATH:SEConds?**

The HP-TIM alarm result is only available when HP-TIM from J1 Trace section has been enabled. HPPLm, selects the HP-PLM (High Order Path - Payload Label Mismatch), declared upon receipt of five consecutive frames with mismatched VC (Virtual Container) signal labels.

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of High Order Path (HOP) alarm.

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:SEConds?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:TYPE AUA* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH ON* FETC:DATA:TEL:SDH:ALAR:HOP:PATH:SEC? AUA Returns the number of seconds of HOP alarm.
Note	For 8120NGE/8130NGE/8130NGEv2 modules, choices are AUAis HPRDi ESD ECD EPD H4LOM AULop HPUNeq HPTim HPPLm. For 8140 module, choices are AUAis HPRDi ESD ECD EPD AULop HPUNeq HPTim HPPLm.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:CURRent?**

Description	<p>This query returns the current status of High Order Path (HOP) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP: PATH:CURRent? <wsp>AUAis HPRDi ESD ECD EPD H4LOM AULop HPUNeq HPTim HPPLm</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AUAis HPRDi ESD ECD EPD H4LOM AULop HPUNeq HPTim HPPLm.</p> <p>Selects the type of HOP (High Order Path) alarm.</p> <p>AUAis, selects the AU-AIS (Administrative Unit - Alarm Indication Signal) when the H1 and H2 bytes contain an all-ones pattern in three consecutive frames.</p> <p>HPRDi, selects the HP-RDI (High Order Path - Remote Defect Indication) alarm, when bits 5, 6 and 7 of the G1 byte contain the "100" or "111" pattern in five consecutive frames.</p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:CURRent?**

ESD, selects the ERDI-SD (Enhanced RDI - Server Defect) when bits 5, 6 and 7 of the G1 byte contain the "101" pattern in five consecutive frames.

ECD, selects the ERDI-CD (Enhanced RDI - Connectivity Defect) alarm, when bits 5, 6 and 7 of the G1 byte contain the "110" pattern in five consecutive frames.

EPD, selects the ERDI-PD (Enhanced RDI - Payload Defect) alarm, when bits 5, 6 and 7 of the G1 byte contain the "010" pattern in five consecutive frames.

H4LOM, selects the H4-LOM (H4 - Loss of Multiframe), for TU structured optical frames, the H4-LOM alarm indicates that the system loss track of the H4 byte multiframe indicator sequence.

AULop, selects the AU-LOP (Administrative Unit - Loss of Pointer), for non-concatenated payloads, the AU-LOP alarm indicates that a valid pointer is not found in N consecutive frames (where $8 < N \leq 10$), or N consecutive NDFs ("1001" pattern) are detected.

HPUNeq, selects the HP-UNEQ (High Order Path - Unequipped) when the C2 bytes contain "00 H" in five consecutive frames.

HPTim, selects the HP-TIM (High Order Path - Trace Identifier Mismatch) which indicates that none of the sampled path trace strings match the expected message value.

:FETCh[1..n]:DATA:TELecom:SDH:ALARm:HOP: PATH:CURRent?

The HP-TIM alarm result is only available when HP-TIM from J1 Trace section has been enabled. HPPLm, selects the HP-PLM (High Order Path - Payload Label Mismatch), declared upon receipt of five consecutive frames with mismatched VC (Virtual Container) signal labels.

Response Syntax <Current>

Response(s)

Current:

The response data syntax for <Current> 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.

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:HOP:
PATH:CURRent?****Example(s)**

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:TYPE
AUA
* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH ON
* FETC:DATA:TEL:SDH:ALAR:HOP:PATH:CURR?
AUA Returns the current alarm status.

Note

For **8120NGE/8130NGE/8130NGEv2** modules,
choices are AUAis | HPRDi | ESD | ECD |
EPD | H4LOM | AULop | HPUNeq | HPTim | HPPLm.
For **8140** module, choices are AUAis | HPRDi |
ESD | ECD | EPD | AULop | HPUNeq | HPTim |
HPPLm.

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:HISTory?**

Description	<p>This query returns the history status of Low Order Path (LOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:HISTory? <wsp>BIP2 LPRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 LPRei.</p> <p>Selects the type of LOP (Low Order Path) error. BIP2, selects the BIP-2 (Bit-Interleave Parity - 2 bits) error which indicates a Low Order Path (LOP) parity error by performing a routine even-parity check over all bytes of the previous VC (Virtual Container) frame.</p> <p>LPRei, selects the LP-REI (Low Order Path Remote Error Indicator) error when bit 3 of the V5 byte is set to "1".</p>
Response Syntax	<p><History></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Low Order Path (LOP) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN:TYPE BIP2* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AMO 15* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:INJ* FETC:DATA:TEL:SDH:ERR:LOP:PATH:HIST? BIP2 Returns the error history status.
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AMOUNT* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ERROr:LOP:
PATH:SEConds?**

Description This query returns the number of seconds within which Low Order Path (LOP) error occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ERROr:LOP:
PATH:SEConds? <wsp>BIP2|LPRei

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2|LPRei.
Selects the type of LOP (Low Order Path) error. BIP2, selects the BIP-2 (Bit-Interleave Parity - 2 bits) error which indicates a Low Order Path (LOP) parity error by performing a routine even-parity check over all bytes of the previous VC frame.
LPRei, selects the LP-REI (Low Order Path Remote Error Indicator) error when bit 3 of the V5 byte is set to "1".

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Low Order Path (LOP) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN:TYPE BIP2 * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AMO 15 * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:INJ * FETC:DATA:TEL:SDH:ERR:LOP:PATH:SEC? BIP2 Returns the number of errored seconds.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AMOUNT * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:INJECT

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:CURRent?**

Description	<p>This query returns the current status of Low Order Path (LOP) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:CURRent?<wsp>BIP2 LPRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 LPRei.</p> <p>Selects the type of LOP (Low Order Path) error. BIP2, selects the BIP-2 (Bit-Interleave Parity - 2 bits) error which indicates a Low Order Path (LOP) parity error by performing a routine even-parity check over all bytes of the previous VC frame.</p> <p>LPRei, selects the LP-REI (Low Order Path Remote Error Indicator) error when bit 3 of the V5 byte is set to "1".</p>
Response Syntax	<p><Current></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of Low Order Path (LOP) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN:TYPE BIP2* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AMO 15* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:INJ* FETC:DATA:TEL:SDH:ERR:LOP:PATH:CURR? BIP2 Returns the current error status.
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AMOUNT* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:INJECT

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:COUNT?**

Description	<p>This query returns the count for Low Order Path error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:COUNT? <wsp>BIP2 LPRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 LPRei.</p> <p>Selects the type of LOP (Low Order Path) error. BIP2, selects the BIP-2 (Bit-Interleave Parity - 2 bits) error which indicates a Low Order Path (LOP) parity error by performing a routine even-parity check over all bytes of the previous VC frame.</p> <p>LPRei, selects the LP-REI (Low Order Path Remote Error Indicator) error when bit 3 of the V5 byte is set to "1".</p>
Response Syntax	<p><Count></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:COUNT?**

Response(s)	Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of Low Order Path (LOP) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN: TYPE BIP2 * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AMO 15 * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:INJ * FETC:DATA:TEL:SDH:ERR:LOP:PATH:COUN? BIP2 Returns the error count.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:AMOUNT * SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:INJect

:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP: PATH:RATE?

Description This query returns the current rate of Low Order Path (LOP) error.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:RATE?<wsp>BIP2|LPRei

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2|LPRei.
Selects the type of LOP (Low Order Path) error.
BIP2, selects the BIP-2 (Bit-Interleave Parity - 2 bits) error which indicates a Low Order Path (LOP) parity error by performing a routine even-parity check over all bytes of the previous VC frame.
LPRei, selects the LP-REI (Low Order Path Remote Error Indicator) error when bit 3 of the V5 byte is set to "1".

Response Syntax <Rate>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:LOP:
PATH:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of Low Order Path (LOP) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:MAN:TYPE BIP2* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:AMO 15* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:INJ* FETC:DATA:TEL:SDH:ERR:LOP:PATH:RATE? BIP2 Returns the error rate.
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:AMOUNT* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP:
PATH:HISTory?**

Description	<p>This query returns the history status of Low Order Path (LOP) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP: PATH:HISTory?<wsp>TUAis TULop LPRDi LPRFi LPTim LPPLm LPUNeq ESD ECD EPD</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TUAis TULop LPRDi LPRFi LPTim LPPLm LPUNeq ESD ECD EPD.</p> <p>Selects the type of LOP (Low Order Path) alarm.</p> <p>TUAis, selects the TU-AIS (Tributary Unit - Alarm Indication Signal) alarm when V1 and V2 bytes for the TU path contain an all-ones pattern in five consecutive superframes.</p> <p>TULop, selects the TU-LOP (Tributary Unit - Loss Of Pointer) alarm, indicates that a valid pointer is not found in N consecutive superframes (where 8 <= N <= 10), or if N consecutive NDFs ("1001" pattern) are detected.</p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP:
PATH:HISTory?**

LPRDi, selects the LP-RDI (Tributary Unit - Remote Defect Indication) alarm when bit 8 of V5 byte contains "1" in five consecutive TU superframes while bits 6 and 7 of the K4 byte contain the "00" or "11" pattern.

LPRFi, selects the LP-RFI (Low Order Path - Remote Failure Indication) (available with VC-11 only) alarm when bit 4 of V5 byte contains "1" in five consecutive superframes.

LPTim, selects the LP-TIM (Low Order Path - Trace Identifier Mismatch), defect indicates that none of the sampled LP trace strings match the expected message value.

The LP-TIM alarm result is only available when LP-TIM from J2 Trace section has been enabled.

LPPLm, selects the LP-PLM (Low Order Path - Payload Label Mismatch), declared upon receipt of five consecutive superframes with mismatched LP Signal (bits 5 through 7 of the V5 byte are "000", "001" or "111")

LPUNeq, selects the LP-UNEQ (Low Order Path - Unequipped), when bit 5 through 7 of the V5 byte contain "000" for five consecutive superframes.

ESD, selects the ERDI-SD (Enhanced RDI - Server Defect) alarm when bits 5, 6, and 7 of the K4 byte contain the "101" pattern, and bit 8 of the V5 byte contain "1", in five consecutive LP superframes.

ECD, selects the ERDI-CD (Enhanced RDI - Connectivity Defect) alarm when bits 5, 6, and 7 of the K4 byte contain the "110" pattern, and bit 8 of the V5 byte contain "1", in five consecutive LP superframes.

:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP: PATH:HISTory?

EPD, selects the ERDI-PD (Enhanced RDI - Path Payload Defect) alarm when bits 5, 6, and 7 of the K4 byte contain the "010" pattern, and bit 8 of the V5 byte contain "0", in five consecutive LP superframes.

Response Syntax <History>

Response(s) History:
The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the history status of Low Order Path (LOP) 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.

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP:
PATH:HISTory?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE TUA* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH ON* FETC:DATA:TEL:SDH:ALAR:LOP:PATH:HIST? TUA Returns the alarm history status.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:LOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:LOP:PATH

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP:
PATH:SEConds?**

Description This query returns the number of seconds within which Low Order Path (LOP) alarm occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP:
PATH:SEConds? <wsp> TUAis|TULop|LPRDi|
LPRFi|LPTim|LPPLm|LPUNeq|ESD|ECD|EPD

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
TUAis|TULop|LPRDi|LPRFi|LPTim|LPPLm|LPUNeq|ESD|ECD|EPD.
Selects the type of LOP (Low Order Path) alarm.
TUAis, selects the TU-AIS (Tributary Unit - Alarm Indication Signal) alarm when V1 and V2 bytes for the TU path contain an all-ones pattern in five consecutive superframes.
TULop, selects the TU-LOP (where $8 \leq N \leq 10$) (Tributary Unit - Loss Of Pointer) alarm, indicates that a valid pointer is not found in N consecutive superframes (where $8 \leq N \leq 10$), or if N consecutive NDFs ("1001" pattern) are detected.

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP:
PATH:SEConds?**

LPRDi, selects the LP-RDI (Tributary Unit - Remote Defect Indication) alarm when bit 8 of V5 byte contains "1" in five consecutive TU superframes while bits 6 and 7 of the K4 byte contain the "00" or "11" pattern.

LPRFi, selects the LP-RFI (Low Order Path - Remote Failure Indication) (available with VC-11 only) alarm when bit 4 of V5 byte contains "1" in five consecutive superframes.

LPTim, selects the LP-TIM (Low Order Path - Trace Identifier Mismatch), defect indicates that none of the sampled LP trace strings match the expected message value. The LP-TIM alarm result is only available when LP-TIM from J2 Trace section has been enabled.

LPPLm, selects the LP-PLM (Low Order Path - Payload Label Mismatch), declared upon receipt of five consecutive superframes with mismatched LP Signal (bits 5 through 7 of the V5 byte are "000", "001" or "111")

LPUNeq, selects the LP-UNEQ (Low Order Path - Unequipped), when bit 5 through 7 of the V5 byte contain "000" for five consecutive superframes.

ESD, selects the ERDI-SD (Enhanced RDI - Server Defect) alarm when bits 5, 6, and 7 of the K4 byte contain the "101" pattern, and bit 8 of the V5 byte contain "1", in five consecutive LP superframes.

ECD, selects the ERDI-CD (Enhanced RDI - Connectivity Defect) alarm when bits 5, 6, and 7 of the K4 byte contain the "110" pattern, and bit 8 of the V5 byte contain "1", in five consecutive LP superframes.

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP:
PATH:SEConds?**

EPD, selects the ERDI-PD (Enhanced RDI - Path Payload Defect) alarm when bits 5, 6, and 7 of the K4 byte contain the "010" pattern, and bit 8 of the V5 byte contain "0", in five consecutive LP superframes.

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Low Order Path (LOP) alarm.

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP:
PATH:SEConds?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE TUA* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH ON* FETC:DATA:TEL:SDH:ALAR:LOP:PATH:SEC? TUA Returns the number of seconds of LOP alarm.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP:
PATH:CURRent?**

Description	<p>This query returns the current status of Low Order Path (LOP) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP: PATH:CURRent?<wsp>TUAis TULop LPRDi LPRFi LPTim LPPLm LPUNeq ESD ECD EPD</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TUAis TULop LPRDi LPRFi LPTim LPPLm LPUNeq ESD ECD EPD.</p> <p>Selects the type of LOP (Low Order Path) alarm.</p> <p>TUAis, selects the TU-AIS (Tributary Unit - Alarm Indication Signal) alarm when V1 and V2 bytes for the TU path contain an all-ones pattern in five consecutive superframes.</p> <p>TULop, selects the TU-LOP (Tributary Unit - Loss of Pointer) alarm(where 8 <= N <= 10), indicates that a valid pointer is not found in N consecutive superframes (where 8 <= N <= 10), or if N consecutive NDFs ("1001" pattern) are detected.</p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP:
PATH:CURRent?**

LPRDi, selects the LP-RDI (Tributary Unit - Remote Defect Indication) alarm when bit 8 of V5 byte contains "1" in five consecutive TU superframes while bits 6 and 7 of the K4 byte contain the "00" or "11" pattern.

LPRFi, selects the LP-RFI (Low Order Path - Remote Failure Indication) (available with VC-11 only) alarm when bit 4 of V5 byte contains "1" in five consecutive superframes.

LPTim, selects the LP-TIM (Low Order Path - Trace Identifier Mismatch), defect indicates that none of the sampled LP trace strings match the expected message value. The LP-TIM alarm result is only available when LP-TIM from J2 Trace section has been enabled.

LPPLm, selects the LP-PLM (Low Order Path - Payload Label Mismatch), declared upon receipt of five consecutive superframes with mismatched LP Signal (bits 5 through 7 of the V5 byte are "000", "001" or "111")

LPUNeq, selects the LP-UNEQ (Low Order Path - Unequipped), when bit 5 through 7 of the V5 byte contain "000" for five consecutive superframes.

ESD, selects the ERDI-SD (Enhanced RDI - Server Defect) alarm when bits 5, 6, and 7 of the K4 byte contain the "101" pattern, and bit 8 of the V5 byte contain "1", in five consecutive LP superframes.

ECD, selects the ERDI-CD (Enhanced RDI - Connectivity Defect) alarm when bits 5, 6, and 7 of the K4 byte contain the "110" pattern, and bit 8 of the V5 byte contain "1", in five consecutive LP superframes.

:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:LOP: PATH:CURRent?

EPD, selects the ERDI-PD (Enhanced RDI - Path Payload Defect) alarm when bits 5, 6, and 7 of the K4 byte contain the "010" pattern, and bit 8 of the V5 byte contain "0", in five consecutive LP superframes.

Response Syntax <Current>

Response(s) Current:
The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the current status of Low Order Path (LOP) 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.

**:FETCh[1..n]:DATA:TELeom:SDH:ALARm:LOP:
PATH:CURREnt?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:TYPE TUA* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH ON* FETC:DATA:TEL:SDH:ALAR:LOP:PATH:CURR? TUA Returns the current alarm status.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELeom:SDH:ALARm: LOP:PATH:TYPE* SOURce[1..n]:DATA:TELeom:SDH:ALARm: LOP:PATH

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:TYPE

Description This command selects the Burst type for the High Order Path (HOP) error.

At *RST, this value is set to BERRor.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:BURSt:TYPE <wsp>BERRor|HPRei

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor HPRei. Selects the Burst type for the High Order Path (HOP) error.</p> <p>BERRor, selects B3 (BERROR) as HOP error.</p> <p>HPRei, selects HPREI (High Order path - Remote Error Indicator) as HOP error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:PATH:BURSt:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:TYPE?**

Description	This query returns the Burst type for the High Order Path (HOP) error. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELecom:SDH:ERRor:
HOP:PATH:BURSt:TYPE?****Response(s)**

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the High Order Path (HOP) error.

BERROR, B3 (BERROR) is selected as HOP error. HPREI, High Order path - Remote Error Indicator (HPREI) is selected as HOP error.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:
TYPE BERR

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:
TYPE? Returns BERROR

See Also

* SOURce[1..n]:DATA:TELecom:SDH:ERRor:
HOP:PATH:BURSt:TYPE

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:MODE

Description This command selects the Burst mode for the High Order Path (HOP) error.

At *RST, this value is set to SINGLE.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:BURSt:MODE<wsp>SINGle|REPeat

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:MODE**

Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat. Selects the Burst mode for the High Order Path (HOP) error.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: MODE SING</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: MODE? Returns SINGLE</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:MODE?**

Description This query returns the Burst mode for the High Order Path (HOP) error.

At *RST, this value is set to SINGLE.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:BURSt:MODE?

Parameter(s) None

Response Syntax <Mode>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:MODE?****Response(s)**

Mode:

The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the High Order Path (HOP) error.

SINGLE, Single is selected as Burst mode.

REPEAT, Repeat is selected as Burst mode.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:
TYPE BERR

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:
MODE SING

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:
MODE? Returns SINGLE

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:MODE

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:DURation

Description

This command sets the duration for the High Order Path (HOP) error.

At *RST, this value is set to 1.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:  
PATH:BURSt:DURation <wsp> <Duration>  
|MAXimum|MINimum
```

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:DURation**

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the High Order Path (HOP) error. Choices are 1 through 14400000.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: MODE SING</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: DUR 15</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: DUR? Returns 15</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:DURation**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:DURation?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:DURation?**

Description	<p>This query returns the duration for the High Order Path (HOP) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:BURSt:DURation?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<pre><Duration></pre>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:DURation?

Response(s)	<p>Duration:</p> <p>The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the duration for the High Order Path (HOP) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: MODE SING</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: DUR 15</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: DUR? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:DURation</p>

**:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:PERiod**

Description This command sets the period for the High Order Path (HOP) error.

At *RST, this value is set to 1.

Syntax :SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:BURSt:PERiod <wsp> <Period>
|MAXimum|MINimum

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:PERiod

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the High Order Path (HOP) error. Choices are 1 through 14400000.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:MODE REP* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:PER 15* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:PER? Returns 15

**:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:
HOP:PATH:BURSt:PERiod**

See Also

* SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:
HOP:PATH:BURSt:TYPE

* SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:
HOP:PATH:BURSt:MODE

* SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:
HOP:PATH:BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERROr:
HOP:PATH:BURSt:PERiod?**

Description	<p>This query returns the period for the High Order Path (HOP) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERROr:HOP: PATH:BURSt:PERiod?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<p><Period></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:PERiod?****Response(s)**

Period:

The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the period for the High Order Path (HOP) error.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:
TYPE BERR

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:
MODE REP

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:
PER 15

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:
PER? Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:PERiod

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt

Description This command enables or disables the Burst for the High Order Path (HOP) error.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP:
PATH:BURSt <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the High Order Path (HOP) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS: TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS ON</p> <p>* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: HOP:PATH:BURSt?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt?**

Description	This query returns the status of Burst for the High Order Path (HOP) error. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:HOP: PATH:BURSt?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Burst for the High Order Path (HOP) error.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS:
TYPE BERR

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS
ON

* SOUR:DATA:TEL:SDH:ERR:HOP:PATH:BURS?
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
HOP:PATH:BURSt

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:TYPE

Description This command selects the Burst type for the High Order Path (HOP) alarm.

At *RST, this value is set to AUAis.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:TYPE<wsp>AUAis | HPRDi |
ESD | ECD | EPD | AULop | HPUNeq

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:TYPE

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

AUAis|HPRDi|ESD|ECD|EPD|AULop|HPUNeq.

Selects the Burst type for the High Order Path (HOP) alarm.

AUAis, selects Administrative Unit - Alarm Indication Signal (AUAIS) as HOP alarm.

HPRDi, selects High Order path - Remote Defect Indication (HPRDI) as HOP alarm.

ESD, selects Enhanced RDI - Server Defect (ESD) as HOP alarm.

ECD, selects Enhanced RDI - Connectivity Defect (ECD) as HOP alarm.

EPD, selects Enhanced RDI - Payload Defect (EPD) as HOP alarm.

AULop, selects Administrative Unit - Loss of Pointer (AULOP) as HOP alarm.

HPUNeq, selects High Order Path - Unequipped (HPUNEQ) as HOP alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:
TYPE AUA

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:
TYPE? Returns AUAIS

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:TYPE**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:TYPE?**

Description	This query returns the Burst type for the High Order Path (HOP) alarm. At *RST, this value is set to AUAis.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:TYPE?**

Response(s)

Alarm:

The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the High Order Path (HOP) alarm.

AUAIS, Administrative Unit - Alarm Indication Signal (AUAIS) is selected as HOP alarm.

HPRDI, High Order path - Remote Defect Indication (HPRDI) is selected as HOP alarm.

ESD, Enhanced RDI - Server Defect (ESD) is selected as HOP alarm.

ECD, Enhanced RDI - Connectivity Defect (ECD) is selected as HOP alarm.

EPD, Enhanced RDI - Payload Defect (EPD) is selected as HOP alarm.

AULOP, Administrative Unit - Loss of Pointer (AULOP) is selected as HOP alarm.

HPUNEQ, High Order Path - Unequipped (HPUNEQ) is selected as HOP alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:
TYPE AUA

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:
TYPE? Returns AUAIS

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:TYPE

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:MODE****Description**

This command selects the Burst mode for the High Order Path (HOP) alarm.

At *RST, this value is set to SINGLE.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:MODE<wsp>SINGle | REPeat

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:MODE**

Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat.</p> <p>Selects the Burst mode for the High Order Path (HOP) alarm.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:TYPE AUA</p> <p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:MODE? Returns SINGLE</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:HOP:PATH:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:MODE?**

Description	This query returns the Burst mode for the High Order Path (HOP) alarm. At *RST, this value is set to SINGLE.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:MODE?
Parameter(s)	None
Response Syntax	<Mode>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:MODE?**

Response(s)	Mode: The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the High Order Path (HOP) alarm. SINGLE, Single is selected as Burst mode. REPEAT, Repeat is selected as Burst mode.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS: TYPE AUA * SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS: MODE SING * SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS: MODE? Returns SINGLE
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:MODE

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:DURation****Description**

This command sets the duration for the High Order Path (HOP) alarm.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:DURation <wsp> <Duration>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:DURation**

Parameter(s) Duration:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the High Order Path (HOP) alarm. Choices are 1 through 14400000.

Example(s) * SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:
TYPE AUA
* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:
MODE SING
* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:
DUR 15
* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:
DUR? Returns 15

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:DURation**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:DURation?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:DURation?**

Description	<p>This query returns the duration for the High Order Path (HOP) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:DURation?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<p><Duration></p>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:DURation?

Response(s)	<p>Duration:</p> <p>The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the duration for the High Order Path (HOP) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:TYPE AUA</p> <p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:DUR 15</p> <p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:DUR? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:HOP:PATH:BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:HOP:PATH:BURSt:DURation</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:PERiod**

Description

This command sets the period for the High Order Path (HOP) alarm.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:PERiod <wsp> <Period>
|MAXimum |MINimum

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:PERiod**

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the High Order Path (HOP) alarm. Choices are 1 through 14400000.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALARm:HOP:PATH: BURS:TYPE AUA* SOUR:DATA:TEL:SDH:ALARm:HOP:PATH: BURS:MODE REP* SOUR:DATA:TEL:SDH:ALARm:HOP:PATH: BURS:PER 15* SOUR:DATA:TEL:SDH:ALARm:HOP:PATH: BURS:PER? Returns 15

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:PERiod**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:PERiod?**

Description	<p>This query returns the period for the High Order Path (HOP) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:PERiod?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p>
Response Syntax	<pre><Period></pre>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:PERiod?**

Response(s)	<p>Period:</p> <p>The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the period for the High Order Path (HOP) alarm.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALARm:HOP:PATH: BURS:TYPE AUA* SOUR:DATA:TEL:SDH:ALARm:HOP:PATH: BURS:MODE REP* SOUR:DATA:TEL:SDH:ALARm:HOP:PATH: BURS:PER 15* SOUR:DATA:TEL:SDH:ALARm:HOP:PATH: BURS:PER? Returns 15
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:MODE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:PERiod

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt****Description**

This command enables or disables the Burst for the High Order Path (HOP) alarm.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt<wsp><Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the High Order Path (HOP) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS: TYPE AUA</p> <p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS ON</p> <p>* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt?**

Description	This query returns the status of Burst for the High Order Path (HOP) alarm. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: HOP:PATH:BURSt?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt?**

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of Burst for the High Order Path (HOP) alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS:
TYPE AUA

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS
ON

* SOUR:DATA:TEL:SDH:ALAR:HOP:PATH:BURS
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
HOP:PATH:BURSt

**:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:
LOP:PATH:BURSt:TYPE**

Description This command selects the Burst type for the Low Order Path (LOP) error.

At *RST, this value is set to BIP2.

Syntax :SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:
LOP:PATH:BURSt:TYPE<wsp>BIP2|LPRei

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BIP2 LPRei.</p> <p>Selects the Burst type for the Low Order Path (LOP) error.</p> <p>BIP2, selects Bit-Interleaved Parity - 2 bits (BIP2) as LOP error.</p> <p>LPRei, selects Low Order Path Remote Error Indication (LPREI) as LOP error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:TYPE BIP2</p> <p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:TYPE? Returns BIP2</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:BURSt:TYPE?</p>

**:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:TYPE?**

Description	This query returns the Burst type for the Low Order Path (LOP) error. At *RST, this value is set to BIP2.
Syntax	:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELecom:SDH:ERRor: LOP:PATH:BURSt:TYPE?

Response(s)

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the Low Order Path (LOP) error.

BIP2, Bit-Interleaved Parity - 2 bits (BIP2) is selected as LOP error.

LPREI, Low Order Path Remote Error Indication (LPREI) is selected as LOP error.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:TYPE BIP2

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:TYPE? Returns BIP2

See Also

* SOURce[1..n]:DATA:TELecom:SDH:ERRor:LOP:PATH:BURSt:TYPE

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:MODE****Description**

This command selects the Burst mode for the Low Order Path (LOP) error.

At *RST, this value is set to SINGLE.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:MODE<wsp>SINGle|REPeat

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:MODE**

Parameter(s) Mode:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle|REPeat.
Selects the Burst mode for the Low Order Path (LOP) error.
SINGle, selects Single as Burst mode.
REPeat, selects Repeat as Burst mode.

Example(s)

- * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:TYPE BIP2
- * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:MODE SING
- * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:MODE? Returns SINGLE

See Also

- * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:BURSt:TYPE
- * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:LOP:PATH:BURSt:MODE?

**:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:MODE?**

Description	This query returns the Burst mode for the Low Order Path (LOP) error. At *RST, this value is set to SINGLE.
Syntax	:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:MODE?
Parameter(s)	None
Response Syntax	<Mode>

**:SOURce[1..n]:DATA:TELecom:SDH:ERRor:
LOP:PATH:BURSt:MODE?**

Response(s)

Mode:

The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the Low Order Path (LOP) error.

SINGLE, Single is selected as Burst mode.

REPEAT, Repeat is selected as Burst mode.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
TYPE BIP2

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
MODE SING

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
MODE? Returns SINGLE

See Also

* SOURce[1..n]:DATA:TELecom:SDH:ERRor:
LOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELecom:SDH:ERRor:
LOP:PATH:BURSt:MODE

**:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:
LOP:PATH:BURSt:DURation****Description**

This command sets the duration for the Low Order Path (LOP) error.

At *RST, this value is set to 1.

Syntax

:SOURCE[1..n]:DATA:TELEcom:SDH:ERROR:
LOP:PATH:BURSt:DURation <wsp> <Duration>
|MAXimum|MINimum

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:DURation

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the Low Order Path (LOP) error. Choices are 1 through 14400000.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:TYPE BIP2* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:MODE SING* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:DUR 15* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:DUR? Returns 15

**:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:DURation**

See Also

* SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:TYPE

* SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:MODE

* SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:DURation?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:DURation?**

Description	<p>This query returns the duration for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:DURation?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<p><Duration></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:DURation?****Response(s)**

Duration:

The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the duration for the Low Order Path (LOP) error.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
TYPE BIP2

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
MODE SING

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
DUR 15

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
DUR? Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:DURation

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:PERiod**

Description

This command sets the period for the Low Order Path (LOP) error.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:PERiod <wsp> <Period>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:PERiod**

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the Low Order Path (LOP) error. Choices are 1 through 14400000.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS: TYPE BIP2</p> <p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS: MODE REP</p> <p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS: PER 15</p> <p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS: PER? Returns 15</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:PERiod**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:PERiod?**

Description	<p>This query returns the period for the Low Order Path (LOP) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt:PERiod?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<p><Period></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:PERiod?**

Response(s) Period:
The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the period for the Low Order Path (LOP) error.

Example(s) * SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
TYPE BIP2
* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
MODE REP
* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
PER 15
* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
PER? Returns 15

See Also * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:MODE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt:PERiod

**:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt**

Description This command enables or disables Burst for the Low Order Path (LOP) error.

At *RST, this value is set to OFF.

Syntax :SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt <wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the Low Order Path (LOP) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS: TYPE BIP2</p> <p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS ON</p> <p>* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt?</p>

**:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt?**

Description	This query returns the status of Burst for the Low Order Path (LOP) error. At *RST, this value is set to OFF.
Syntax	:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: LOP:PATH:BURSt?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of Burst for the Low Order Path (LOP) error.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS:
TYPE BIP2

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS
ON

* SOUR:DATA:TEL:SDH:ERR:LOP:PATH:BURS?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
LOP:PATH:BURSt

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE****Description**

This command selects the Burst type for the Low Order Path (LOP) alarm.

At *RST, this value is set to AUAis.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:  
LOP:PATH:BURSt:TYPE<wsp>AUAis|LPRDi|  
ESD|ECD|EPD|LPRFi|TULop|LPUNeq
```

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:TYPE

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

AUAis|LPRDi|ESD|ECD|EPD|LPRFi|TULop|LPUNeq.

Selects the Burst type for the Low Order Path (LOP) alarm.

AUAis, selects Administrative Unit - Alarm Indication Signal (AUAIS) as LOP alarm.

LPRDi, selects Low Order Path - Remote Defect Indication (LPRDI) as LOP alarm.

ESD, selects Enhanced RDI - Server Defect (ESD) as LOP alarm.

ECD, selects Enhanced RDI - Connectivity Defect (ECD) as LOP alarm.

EPD, selects Enhanced RDI - Payload Defect (EPD) as LOP alarm.

LPRFi, selects Low Order Path - Remote Failure Indication (LPRFI) as LOP alarm.

TULop, selects Tributary Unit - Loss Of Pointer (TULOP) as LOP alarm.

LPUNeq, selects Low Order Path - Unequipped (LPUNEQ) as LOP alarm.

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE**

Example(s)	* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS: TYPE AUA * SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS: TYPE? Returns AUAIS
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE?**

Description This query returns the Burst type for the Low Order Path (LOP) alarm.

At *RST, this value is set to AUAis.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE?

Parameter(s) None

Response Syntax <Alarm>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE?**

Response(s)

Alarm:

The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the Low Order Path (LOP) alarm.

AUAIS, Administrative Unit - Alarm Indication Signal (AUAIS) is selected as LOP alarm.

LPRDI, Low Order Path - Remote Defect Indication (LPRDI) is selected as LOP alarm.

ESD, Enhanced RDI - Server Defect (ESD) is selected as LOP alarm.

ECD, Enhanced RDI - Connectivity Defect (ECD) is selected as LOP alarm.

EPD, Enhanced RDI - Payload Defect (EPD) is selected as LOP alarm.

LPRFI, Low Order Path - Remote Failure Indication (LPRFI) is selected as LOP alarm.

TULOP, Tributary Unit - Loss Of Pointer (TULOP) is selected as LOP alarm.

LPUNEQ, Low Order Path - Unequipped (LPUNEQ) is selected as LOP alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
TYPE AUA

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
TYPE? Returns AUAIS

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE?**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:MODE****Description**

This command selects the Burst mode for the Low Order Path (LOP) alarm.

At *RST, this value is set to SINGLE.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:MODE<wsp>SINGle|REPeat

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:MODE**

Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat.</p> <p>Selects the Burst mode for the Low Order Path (LOP) alarm.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:TYPE AUA</p> <p>* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:MODE? Returns SINGLE</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:LOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:LOP:PATH:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:MODE?**

Description	This query returns the Burst mode for the Low Order Path (LOP) alarm. At *RST, this value is set to SINGLE.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:MODE?
Parameter(s)	None
Response Syntax	<Mode>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:MODE?

Response(s)

Mode:

The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the Low Order Path (LOP) alarm.

SINGLE, Single is selected as Burst mode.

REPEAT, Repeat is selected as Burst mode.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
TYPE AUA

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
MODE SING

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
MODE? Returns SINGLE

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:MODE

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:DURation****Description**

This command sets the duration for the Low Order Path (LOP) alarm.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:DURation <wsp> <Duration>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:DURation**

Parameter(s) Duration:
The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the Low Order Path (LOP) alarm. Choices are 1 through 14400000.

Example(s) * SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
TYPE AIS
* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
MODE SING
* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
DUR 15
* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
DUR? Returns 15

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:DURation**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:DURation?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:DURation?**

Description	<p>This query returns the duration for the Low Order Path (LOP) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:DURation?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<p><Duration></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:DURation?**

Response(s)	Duration: The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the duration for the Low Order Path (LOP) alarm.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS: TYPE AIS * SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS: MODE SING * SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS: DUR 15 * SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS: DUR? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:DURation

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:PERiod**

Description

This command sets the period for the Low Order Path (LOP) alarm.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:PERiod <wsp> <Period>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:PERiod**

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the Low Order Path (LOP) alarm. Choices are 1 through 14400000.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:TYPE AUA* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:MODE REP* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:PER 15* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:PER? Returns 15

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:PERiod**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:PERiod?**

Description	<p>This query returns the period for the Low Order Path (LOP) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:PERiod? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<pre><Period></pre>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:PERiod?

Response(s)	<p>Period:</p> <p>The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the period for the Low Order Path (LOP) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS: TYPE AUA</p> <p>* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS: MODE REP</p> <p>* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS: PER 15</p> <p>* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS: PER? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt:PERiod</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt**

Description This command enables or disables Burst for the Low Order Path (LOP) alarm.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt <wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt

Parameter(s)

Set:

The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the Low Order Path (LOP) alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
TYPE AIS

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS
ON

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS?
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt?**

Description	This query returns the status of Burst for the Low Order Path (LOP) alarm. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm: LOP:PATH:BURSt?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of Burst for the Low Order Path (LOP) alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS:
TYPE AIS

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS
ON

* SOUR:DATA:TEL:SDH:ALAR:LOP:PATH:BURS
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
LOP:PATH:BURSt

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
TYPE**

Description	<p>This command selects the type of Regenerator Section (RS) alarm.</p> <p>At *RST, this value is set to LOF1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: TYPE<wsp>LOF1 OOF2</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF1 OOF2.</p> <p>Selects the type of Regenerator Section (RS) alarm.</p> <p>LOF1, selects the Loss of Frame (LOF) which generates the errors in all FAS bits continuously.</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: TYPE

OOF2, selects the Out of Frame (OOF) which generates the errors in all FAS bits for 5 consecutive OTU frames repetitively.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:RS:TYPE LOF1
* SOUR:DATA:TEL:SDH:ALAR:RS:TYPE?
Returns LOF1

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
TYPE?**

Description	This query returns the type of Regenerator Section (RS) alarm. At *RST, this value is set to LOF1.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: TYPE?
Parameter(s)	None
Response Syntax	<Alarm>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: TYPE?

Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Regenerator Section (RS) alarm.</p> <p>LOF1, Loss of Frame (LOF) is selected as Regenerator Section (RS) alarm.</p> <p>OOF2, Out of Frame (OOF) is selected as Regenerator Section (RS) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:RS:TYPE LOF1</p> <p>* SOUR:DATA:TEL:SDH:ALAR:RS:TYPE?</p> <p>Returns LOF1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:TYPE</p>

:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS

Description	<p>This command enables or disables the status of Regenerator Section (RS) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the RS (Regenerator Section) alarm generation.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SDH:ALAR:RS:TYPE LOF1 * SOUR:DATA:TEL:SDH:ALAR:RS ON * SOUR:DATA:TEL:SDH:ALAR:RS? Returns 1</pre>
See Also	<pre>* SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:RS: TYPE * SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: RS?</pre>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS?

Description	This query returns the status of Regenerator Section (RS) alarm generation. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Regenerator Section (RS) alarm generation.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:RS:TYPE LOF1 * SOUR:DATA:TEL:SDH:ALAR:RS ON * SOUR:DATA:TEL:SDH:ALAR:RS? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE

Description	<p>This command selects the manual type of Regenerator Section (RS) error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE<wsp>BERRor FAS
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS. Selects the type of Regenerator Section (RS) error.</p> <p>BERRor, selects the type of Regenerator Section (RS) error as B1.</p> <p>FAS, selects the type of Regenerator Section (RS) error as Frame Alignment Signal (FAS).</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE?</p> <p>Returns BERROR</p>
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
MANual:TYPE?**

Description	This query returns the manual type of Regenerator Section (RS) error. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
MANual:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Regenerator Section (RS) error.</p> <p>BERROR, B1 is selected as Regenerator Section (RS) error.</p> <p>FAS, Frame Alignment Signal (FAS) is selected as Regenerator Section (RS) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE?</p> <p>Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AMOut**

Description	<p>This command sets the amount of Regenerator Section (RS) error to be injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: AMOut<wsp><Amount> MAXimum MINimum</p>
Parameter(s)	<p>Amount: The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AMOut**

Sets the amount of RS (Regenerator Section) error.

Choices are 1 through 50.

Example(s)

- * SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE BERR
- * SOUR:DATA:TEL:SDH:ERR:RS:AMO 15
- * SOUR:DATA:TEL:SDH:ERR:RS:AMO? Returns 15

See Also

- * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE
 - * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AMOut?
-

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AMOUnt?**

Description	<p>This query returns the amount of Regenerator Section (RS) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: AMOUnt? [<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<p><Amount></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AMOUnt?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of Regenerator Section (RS) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:RS:AMO 15 * SOUR:DATA:TEL:SDH:ERR:RS:AMO? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AMOUnt

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: INJect

Description	<p>This command injects the type of Regenerator Section (RS) error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: INJect</pre>
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:RS:AMO 15* SOUR:DATA:TEL:SDH:ERR:RS:INJ
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AMOUNT

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated:TYPE**

Description	<p>This command selects the type of Regenerator Section (RS) error for automated injection.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:TYPE<wsp>BERRor</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter is: BERRor.</p> <p>Selects the type of Regenerator Section (RS) error for automated injection.</p> <p>BERRor, selects the type of line error as B1.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SDH:ERR:RS:AUT:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:RS:AUT:TYPE? Returns BERROR</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor: RS:AUTomated:TYPE? * SOURce[1..n]:DATA:TELEcom:SDH:ERRor: RS:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor: RS:AUTomated</pre>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:TYPE?

Description	This query returns the type of Regenerator Section (RS) error for automated injection. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of Regenerator Section (RS) error for the automated injection.</p> <p>BERROR, B1 is selected as Regenerator Section (RS) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:RS:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:RS:AUT:TYPE?</p> <p>Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:RATE

Description	<p>This command sets the injection rate for the selected Regenerator Section (RS) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:RATE<wsp> <Rate>MAXimum MINimum</p>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated:RATE**

Sets the injection rate for the selected Regenerator Section (RS) error.

Example(s)

```
* SOUR:DATA:TEL:SDH:ERR:RS:AUT:TYPE BERR
* SOUR:DATA:TEL:SDH:ERR:RS:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:SDH:ERR:RS:AUT:RATE?
Returns 1.0E-10
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
RS:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
RS:AUTomated:RATE?
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
RS:AUTomated
```

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Regenerator Section (RS) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:RATE?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injection rate will be returned.</p>
Response Syntax	<p><Rate></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Regenerator Section (RS) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:RS:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:RS:AUT:RATE 1.0E-10</p> <p>* SOUR:DATA:TEL:SDH:ERR:RS:AUT:RATE?</p> <p>Returns 1.0E-10</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated**

Description

This command enables or disables the selected automated Regenerator Section (RS) error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated Regenerator Section (RS) error injection.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:RS:AUT:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:RS:AUT:RATE 1.0E-10* SOUR:DATA:TEL:SDH:ERR:RS:AUT ON* SOUR:DATA:TEL:SDH:ERR:RS:AUT? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated?**

Description	This query returns the status of automated Regenerator Section (RS) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated?
Parameter(s)	None
Response Syntax	<Set>

**:SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated Regenerator Section (RS) error injection.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:RS:AUT:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:RS:AUT:RATE 1.0E-10 * SOUR:DATA:TEL:SDH:ERR:RS:AUT ON * SOUR:DATA:TEL:SDH:ERR:RS:AUT? Returns 1
See Also	* SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated:TYPE * SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated:RATE * SOURCE[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated:CONTInuous**

Description

This command enables or disables the continuous rate of automated Regenerator Section (RS) error injection.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated:CONTInuous<wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated:CONTInuous**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated Regenerator Section (RS) error injection continuously.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:RS:AUT:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:RS:AUT:CONT ON* SOUR:DATA:TEL:SDH:ERR:RS:AUT:CONT? Returns 1 <ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:RS:AUT ON
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated:CONTInuous?**

Description	This query returns the status of continuous rate of automated Regenerator Section (RS) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AUTomated:CONTInuous?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuous rate of automated Regenerator Section (RS) error injection.

Example(s)

```
* SOUR:DATA:TEL:SDH:ERR:RS:AUT:TYPE BERR
* SOUR:DATA:TEL:SDH:ERR:RS:AUT:CONT ON
* SOUR:DATA:TEL:SDH:ERR:RS:AUT:CONT?
Returns 1
* SOUR:DATA:TEL:SDH:ERR:RS:AUT ON
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
RS:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
RS:AUTomated
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
RS:AUTomated:CONTInuous
```

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
TYPE**

Description	<p>This command selects the type of Multiplex Section (MS) alarm.</p> <p>At *RST, this value is set to MSAis.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: TYPE<wsp>MSAis MSRDi</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MSAis MSRDi.</p> <p>Selects the type of MS (Multiplex Section) alarm. MSAis, MS-AIS (Multiplex Section - Alarm Indication Signal) generates "111" pattern for the bits 6, 7 and 8 of the K2 byte.</p>

**:SOURce[1..n]:DATA:TELecom:SDH:ALARm:MS:
TYPE**

MSRDi, MS-RDI (Multiplex Section - Remote Defect Indication) generates "110" pattern for the bits 6, 7 and 8 of the K2 byte.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:MS:TYPE MSA
* SOUR:DATA:TEL:SDH:ALAR:MS:TYPE?
Returns MSAIS

See Also

* SOURce[1..n]:DATA:TELecom:SDH:ALARm:MS:TYPE?

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:TYPE?

Description	This query returns the type of Multiplex Section (MS) alarm. At *RST, this value is set to MSAis.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	Alarm: The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Multiplex Section (MS) alarm. MSAIS, MS-AIS is selected as Multiplex Section (MS) alarm.

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
TYPE?**

MSRDI, MS-RDI is selected as Multiplex Section (MS) alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:MS:TYPE MSA
* SOUR:DATA:TEL:SDH:ALAR:MS:TYPE?
Returns MSAIS

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
MS:TYPE

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS

Description This command enables or disables the status of Multiplex Section (MS) alarm generation.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS
<wsp> <Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the MS (Multiplex Section) alarm generation.

Example(s)

- * SOUR:DATA:TEL:SDH:ALAR:MS:TYPE MSA
- * SOUR:DATA:TEL:SDH:ALAR:MS ON
- * SOUR:DATA:TEL:SDH:ALAR:MS? Returns 1

See Also

- * SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:TYPE
- * SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS?

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS?

Description	<p>This query returns the status of Multiplex Section (MS) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of Multiplex Section (MS) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:MS:TYPE MSA* SOUR:DATA:TEL:SDH:ALAR:MS ON* SOUR:DATA:TEL:SDH:ALAR:MS? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
MANual:TYPE**

Description	<p>This command selects the manual type of Multiplex Section (MS) error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: MANual:TYPE<wsp>BERRor MSRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor MSRei.</p> <p>Selects the type of Multiplex Section (MS) error. BERRor, selects the type of MS error as B2. MSRei, selects the type of MS-REI (Multiplex Section - Remote Error Indication).</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: MANual:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
MANual:TYPE?**

Description	This query returns the manual type of Multiplex Section (MS) error. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: MANual:TYPE?

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Multiplex Section (MS) error. BERROR, B2 is selected as Multiplex Section (MS) error.</p> <p>MSREI, Multiplex Section - Remote Error Indication (MS-REI) is selected as Multiplex Section (MS) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE?</p> <p>Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:MANual:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AMOUnt**

Description	<p>This command sets the amount of Multiplex Section (MS) error to be injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AMOUnt<wsp><Amount> MAXimum MINimum</pre>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AMOUNT**

Sets the amount of MS (Multiplex Section) error.
Choices are 1 through 50.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE BERR
* SOUR:DATA:TEL:SDH:ERR:MS:AMO 15
* SOUR:DATA:TEL:SDH:ERR:MS:AMO?
Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AMOUNT?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AMOUnt?**

Description	<p>This query returns the amount of Multiplex Section (MS) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AMOUnt? [<wsp>MAXimum MINimum]</code>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<code><Amount></code>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AMOUNT?

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of Multiplex Section (MS) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:MS:AMO 15 * SOUR:DATA:TEL:SDH:ERR:MS:AMO? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AMOUNT

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
INJect**

Description	<p>This command injects the type of Multiplex Section (MS) error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:MS:AMO 15 * SOUR:DATA:TEL:SDH:ERR:MS:INJ
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AMOUNT

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS:
HISTory?**

Description	<p>This query returns the history status of Regenerator Section (RS) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS: HISTory? <wsp>LOF1 OOF RSTim</p>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF1 OOF RSTim. Selects the type of RS (Regenerator Section) alarm. LOF1, selects the LOF (Loss of Frame) which generates errors in all the FAS bits continuously. OOF, selects the OOF (Out of Frame) which generates error in all the FAS bits for 5 consecutive OTU frames repetitively. RSTim, selects the RS-TIM (Regenerator Section - Trace Identifier Mismatch) which indicates that none of the sampled regenerator section trace strings match the expected message value.</p>
Response Syntax	<p><History></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS:
HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Regenerator Section (RS) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:RS:TYPE LOF1* SOUR:DATA:TEL:SDH:ALAR:RS ON* FETC:DATA:TEL:SDH:ALAR:RS:HIST? LOF1 <p>Returns the alarm history status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS:
SECOnds?**

Description This query returns the number of seconds within which Regenerator Section (RS) alarm occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS:
SECOnds?<wsp>LOF1|OOF|RSTim

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
LOF1|OOF|RSTim.
Selects the type of RS (Regenerator Section) alarm.
LOF1, selects the LOF (Loss of Frame) which generates errors in all the FAS bits continuously.
OOF, selects the OOF (Out of Frame) which generates error in all the FAS bits for 5 consecutive OTU frames repetitively.
RSTim, selects the RS-TIM (Regenerator Section - Trace Identifier Mismatch) which indicates that none of the sampled regenerator section trace strings match the expected message value.

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS:
SECOnds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Regenerator Section (RS) alarm.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:RS:TYPE LOF1 * SOUR:DATA:TEL:SDH:ALAR:RS ON * FETC:DATA:TEL:SDH:ALAR:RS:SEC? LOF1 Returns the number of seconds of RS alarm.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS:
CURRent?**

Description	<p>This query returns the current status of Regenerator Section (RS) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS: CURRent? <wsp>LOF1 OOF RSTim</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF1 OOF RSTim.</p> <p>Selects the type of RS (Regenerator Section) alarm.</p> <p>LOF1, selects the LOF (Loss of Frame) which generates errors in all the FAS bits continuously.</p> <p>OOF, selects the OOF (Out of Frame) which generates error in all the FAS bits for 5 consecutive OTU frames repetitively.</p> <p>RSTim, selects the RS-TIM (Regenerator Section - Trace Identifier Mismatch) which indicates that none of the sampled regenerator section trace strings match the expected message value.</p>
Response Syntax	<p><Current></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:RS:
CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of Regenerator Section (RS) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:RS:TYPE LOF1* SOUR:DATA:TEL:SDH:ALAR:RS ON* FETC:DATA:TEL:SDH:ALAR:RS:CURR? LOF1 <p>Returns the current alarm status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:
HISTory?**

Description	<p>This query returns the history status of Regenerator Section (RS) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS: HISTory? <wsp>BERRor FAS</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS.</p> <p>Selects the type of RS (Regenerator Section) error.</p> <p>BERRor, selects the type of RS (Regenerator Section) error as B1.</p> <p>FAS, selects the type of Regenerator Section (RS) error as Frame Alignment Signal (FAS).</p>
Response Syntax	<p><History></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:
HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Regenerator Section (RS) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:RS:AMO 15* SOUR:DATA:TEL:SDH:ERR:RS:INJ* FETC:DATA:TEL:SDH:ERR:RS:HIST? BERR <p>Returns the error history status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AMOUnt* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:
SECOnds?**

Description	<p>This query returns the number of seconds within which Regenerator Section (RS) error occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS: SECOnds?<wsp>BERRor FAS</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS. Selects the type of RS (Regenerator Section) error.</p> <p>BERRor, selects the type of RS (Regenerator Section) error as B1.</p> <p>FAS, selects the type of Regenerator Section (RS) error as Frame Alignment Signal (FAS).</p>
Response Syntax	<p><Seconds></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:
SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Regenerator Section (RS) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:RS:AMO 15 * SOUR:DATA:TEL:SDH:ERR:RS:INJ * FETC:DATA:TEL:SDH:ERR:RS:SEC? BERR Returns the number of errored seconds.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AMOUNT * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:INJECT

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:
CURRent?**

Description	<p>This query returns the current status of Regenerator Section (RS) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS: CURRent? <wsp>BERRor FAS</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS.</p> <p>Selects the type of RS (Regenerator Section) error.</p> <p>BERRor, selects the type of RS (Regenerator Section) error as B1.</p> <p>FAS, selects the type of Regenerator Section (RS) error as Frame Alignment Signal (FAS).</p>
Response Syntax	<p><Current></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:
CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of Regenerator Section (RS) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:RS:AMO 15* SOUR:DATA:TEL:SDH:ERR:RS:INJ* FETC:DATA:TEL:SDH:ERR:RS:CURR? BERR <p>Returns the current error status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AMOUnt* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:
COUNT?**

Description	<p>This query returns the count of Regenerator Section (RS) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS: COUNT?<wsp>BERRor FAS</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS.</p> <p>Selects the type of RS (Regenerator Section) error.</p> <p>BERRor, selects the type of RS (Regenerator Section) error as B1.</p> <p>FAS, selects the type of Regenerator Section (RS) error as Frame Alignment Signal (FAS).</p>
Response Syntax	<p><Count></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:
COUNT?**

Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the count of Regenerator Section (RS) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:RS:AMO 15* SOUR:DATA:TEL:SDH:ERR:RS:INJ* FETC:DATA:TEL:SDH:ERR:RS:COUN? BERR <p>Returns the error count.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:AMOUNT* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:
RATE?**

Description	<p>This query returns the current rate of Regenerator Section (RS) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:RATE? <wsp>BERRor FAS</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS.</p> <p>Selects the type of RS (Regenerator Section) error.</p> <p>BERRor, selects the type of RS (Regenerator Section) error as B1.</p> <p>FAS, selects the type of Regenerator Section (RS) error as Frame Alignment Signal (FAS).</p>
Response Syntax	<p><Rate></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:RS:
RATE?****Response(s)****Rate:**

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the current rate of Regenerator Section (RS) error.

Example(s)

```
* SOUR:DATA:TEL:SDH:ERR:RS:MAN:TYPE BERR
* SOUR:DATA:TEL:SDH:ERR:RS:AMO 15
* SOUR:DATA:TEL:SDH:ERR:RS:INJ
* FETC:DATA:TEL:SDH:ERR:RS:RATE? BERR
```

Returns the error rate.

See Also

```
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
AMOUnt
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
INJect
```

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS:
HISTory?**

Description This query returns the history status of Multiplex Section (MS) alarm.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS:
HISTory? <wsp>MSAis|MSRDi

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MSAis|MSRDi.
Selects the type of MS (Multiplex Section) alarm.
MSAis, selects the MS-AIS (Multiplex Section - Alarm Indication Signal) which generates "111" pattern for the bits 6, 7 and 8 of the K2 byte.
MSRDi, selects the MS-RDI (Multiplex Section - Remote Defect Indication) which generates "110" pattern for the bits 6, 7 and 8 of the K2 byte.

Response Syntax <History>

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS:
HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Multiplex Section (MS) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:MS:TYPE MSA* SOUR:DATA:TEL:SDH:ALAR:MS ON* FETC:DATA:TEL:SDH:ALAR:MS:HIST? MSA <p>Returns the alarm history status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS:
SECOnds?**

Description This query returns the number of seconds within which Multiplex Section (MS) alarm occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS:
SECOnds?<wsp>MSAis|MSRDi

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MSAis|MSRDi.
Selects the type of MS (Multiplex Section) alarm.
MSAis, selects the MS-AIS (Multiplex Section - Alarm Indication Signal) which generates "111" pattern for the bits 6, 7 and 8 of the K2 byte.
MSRDi, selects the MS-RDI (Multiplex Section - Remote Defect Indication) which generates "110" pattern for the bits 6, 7 and 8 of the K2 byte.

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELecom:SDH:ALARm:MS:
SECOnd[s?]**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Multiplex Section (MS) alarm.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:MS:TYPE MSA * SOUR:DATA:TEL:SDH:ALAR:MS ON * FETC:DATA:TEL:SDH:ALAR:MS:SEC? MSA Returns the number of seconds of MS alarm.
See Also	* SOURce[1..n]:DATA:TELecom:SDH:ALARm:MS:TYPE * SOURce[1..n]:DATA:TELecom:SDH:ALARm:MS

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS:
CURRent?**

Description This query returns the current status of Multiplex Section (MS) alarm.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS:
CURRent? <wsp>MSAis|MSRDi

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MSAis|MSRDi.
Selects the type of MS (Multiplex Section) alarm.
MSAis, selects the MS-AIS (Multiplex Section - Alarm Indication Signal) which generates "111" pattern for the bits 6, 7 and 8 of the K2 byte.
MSRDi, selects the MS-RDI (Multiplex Section - Remote Defect Indication) which generates "110" pattern for the bits 6, 7 and 8 of the K2 byte.

Response Syntax <Current>

**:FETCh[1..n]:DATA:TELEcom:SDH:ALARm:MS:
CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of Multiplex Section (MS) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:MS:TYPE MSA* SOUR:DATA:TEL:SDH:ALAR:MS ON* FETC:DATA:TEL:SDH:ALAR:MS:CURR? MSA <p>Returns the current status of alarm.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
HISTory?**

Description	<p>This query returns the history status of Multiplex Section (MS) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS: HISTory? <wsp>BERRor MSRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor MSRei. Selects the type of MS (Multiplex Section) error. BERRor, selects the type of MS (Multiplex Section) error as B2. MSRei, selects the type of MS (Multiplex Section) error as MS-REI (Remote Defect Indication Line).</p>
Response Syntax	<p><History></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Multiplex Section (MS) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:MS:AMO 15* SOUR:DATA:TEL:SDH:ERR:MS:INJ* FETC:DATA:TEL:SDH:ERR:MS:HIST? BERR <p>Returns the error history status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AMOUnt* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
SEConds?**

Description	<p>This query returns the number of seconds within which Multiplex Section (MS) error occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS: SEConds?<wsp>BERRor MSRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor MSRei.</p> <p>Selects the type of MS (Multiplex Section) error. BERRor, selects the type of MS (Multiplex Section) error as B2. MSRei, selects the type of MS (Multiplex Section) error as MS-REI (Remote Defect Indication Line).</p>
Response Syntax	<p><Seconds></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Multiplex Section (MS) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:MS:AMO 15 * SOUR:DATA:TEL:SDH:ERR:MS:INJ * FETC:DATA:TEL:SDH:ERR:MS:SEC? BERR Returns the number of errored seconds.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AMOUNT * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:INJECT

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
CURRent?**

Description This query returns the current status of Multiplex Section (MS) error.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
CURRent? <wsp>BERRor|MSRei.

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
BERRor|MSRei.
Selects the type of MS (Multiplex Section) error.
BERRor, selects the type of MS (Multiplex Section) error as B2.
MSRei, selects the type of MS (Multiplex Section) error as MS-REI (Remote Defect Indication Line).

Response Syntax <Current>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of Multiplex Section (MS) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:MS:AMO 15* SOUR:DATA:TEL:SDH:ERR:MS:INJ* FETC:DATA:TEL:SDH:ERR:MS:CURR? BERR <p>Returns the current error status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AMOUnt* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
COUNT?**

Description This query returns the count of Multiplex Section (MS) error.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
COUNT?<wsp>BERRor|MSRei.

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
BERRor|MSRei.
Selects the type of MS (Multiplex Section) error.
BERRor, selects the type of MS (Multiplex Section) error as B2.
MSRei, selects the type of MS (Multiplex Section) error as MS-REI (Remote Defect Indication Line).

Response Syntax <Count>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
COUNT?**

Response(s)	Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of Multiplex Section (MS) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:MS:AMO 15 * SOUR:DATA:TEL:SDH:ERR:MS:INJ * FETC:DATA:TEL:SDH:ERR:MS:COUN? BERR Returns the error count.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AMOUNT * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:INJect

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
RATE?**

Description	<p>This query returns the current rate of Multiplex Section (MS) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:RATE? <wsp>BERRor MSRei.</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor MSRei.</p> <p>Selects the type of MS (Multiplex Section) error. BERRor, selects the type of MS (Multiplex Section) error as B2. MSRei, selects the type of MS (Multiplex Section) error as MS-REI (Remote Defect Indication Line).</p>
Response Syntax	<p><Rate></p>

**:FETCh[1..n]:DATA:TELEcom:SDH:ERRor:MS:
RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of Multiplex Section (MS) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:MS:MAN:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:MS:AMO 15* SOUR:DATA:TEL:SDH:ERR:MS:INJ* FETC:DATA:TEL:SDH:ERR:MS:RATE? BERR <p>Returns the error rate.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AMOUNT* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:INJect

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:TYPE

Description	<p>This command selects the type of Multiplex Section (MS) error for automated injection.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:TYPE<wsp>BERRor MSRei</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor MSRei. Selects the type of Multiplex Section (MS) error for automated injection.</p> <p>BERRor, selects the type of Multiplex Section (MS) error as B2.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated:TYPE**

MSRei, selects the type of MS-REI (Multiplex Section - Remote Error Indication).

Example(s)

* SOUR:DATA:TEL:SDH:ERR:MS:AUT:TYPE BERR
* SOUR:DATA:TEL:SDH:ERR:MS:AUT:TYPE?
Returns BERROR

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
MS:AUTomated:TYPE?
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
MS:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
MS:AUTomated

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:TYPE?

Description	This query returns the type of Multiplex Section (MS) error for automated injection. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of Multiplex Section (MS) error for the automated injection.</p> <p>BERROR, B2 is selected as Multiplex Section (MS) error.</p> <p>MSREI, Multiplex Section - Remote Error Indication (MS-REI) is selected as Multiplex Section (MS) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:MS:AUT:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:MS:AUT:TYPE?</p> <p>Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated:RATE**

Description	<p>This command sets the injection rate for the selected Multiplex Section (MS) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:RATE<wsp> <Rate>MAXimum MINimum</p>
Parameter(s)	<p>Rate: The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated:RATE**

Sets the injection rate for the selected Multiplex Section (MS) error.

Example(s)

```
* SOUR:DATA:TEL:SDH:ERR:MS:AUT:TYPE BERR
* SOUR:DATA:TEL:SDH:ERR:MS:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:SDH:ERR:MS:AUT:RATE?
Returns 1.0E-10
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
MS:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
MS:AUTomated:RATE?
* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
MS:AUTomated
```

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Multiplex Section (MS) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:RATE? <wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injected rate will be returned.</p>
Response Syntax	<p><Rate></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Multiplex Section (MS) error.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SON:ERR:MS:AUT:TYPE BERR * SOUR:DATA:TEL:SON:ERR:MS:AUT:RATE 1.0E-10 * SOUR:DATA:TEL:SON:ERR:MS:AUT:RATE? Returns 1.0E-10</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SONet:ERRor: MS:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: MS:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:SONet:ERRor: MS:AUTomated</pre>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated

Description This command enables or disables the selected automated Multiplex Section (MS) error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated Multiplex Section (MS) error injection.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:MS:AUT:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:MS:AUT:RATE 1.0E-10* SOUR:DATA:TEL:SDH:ERR:MS:AUT ON* SOUR:DATA:TEL:SDH:ERR:MS:AUT? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated?

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated?

Description	This query returns the status of automated Multiplex Section (MS) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated Multiplex Section (MS) error injection.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:MS:AUT:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:MS:AUT:RATE 1.0E-10 * SOUR:DATA:TEL:SDH:ERR:MS:AUT ON * SOUR:DATA:TEL:SDH:ERR:MS:AUT? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:CONTInuous

Description This command enables or disables the continuous rate of automated Multiplex Section (MS) error injection.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated:CONTInuous<wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated:CONTInuous**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated Multiplex Section (MS) error injection continuously.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:MS:AUT:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:MS:AUT:CONT ON* SOUR:DATA:TEL:SDH:ERR:MS:AUT:CONT? Returns 1 <ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:MS:AUT ON
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated:CONTInuous?**

Description	This query returns the status of continuous rate of automated Multiplex Section (MS) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
AUTomated:CONTInuous?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuous rate of automated Multiplex Section (MS) error injection.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:MS:AUT:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:MS:AUT:CONT ON * SOUR:DATA:TEL:SDH:ERR:MS:AUT:CONT? Returns 1 * SOUR:DATA:TEL:SDH:ERR:MS:AUT ON
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: AUTomated:CONTInuous

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor: RS:BURSt:TYPE

Description

This command selects the Burst type for Regenerator Section (RS) error.

At *RST, this value is set to BERRor.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
RS:BURSt:TYPE<wsp>BERRor|FAS

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
RS:BURSt:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor FAS.</p> <p>Selects the Burst type for Regenerator Section (RS) error.</p> <p>BERRor, selects B1 (BERROR) as RS error.</p> <p>FAS, selects Frame Alignment Signal (FAS) as RS error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:TYPE?</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:BURSt:TYPE?

Description	This query returns the Burst type for the Regenerator Section (RS) error. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:
RS:BURSt:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the Regenerator Section (RS) error.</p> <p>BERRor, B1 (BERROR) is selected as RS error.</p> <p>FAS, Frame Alignment Signal (FAS) is selected as RS error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:TYPE</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:MODE

Description This command selects the Burst mode for the Regenerator Section (RS) error.

At *RST, this value is set to SINGLE.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:MODE<wsp>SINGle|REPeat

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:MODE**

Parameter(s)	Mode: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat. Selects the Burst mode for the Regenerator Section (RS) error. SINGle, selects Single as Burst mode. REPeat, selects Repeat as Burst mode.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:RS:BURS:MODE SING * SOUR:DATA:TEL:SDH:ERR:RS:BURS:MODE? Returns SINGLE
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:MODE?

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:MODE?

Description	This query returns the Burst mode for the Regenerator Section (RS) error. At *RST, this value is set to SINGLE.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:MODE?
Parameter(s)	None
Response Syntax	<Mode>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:MODE?****Response(s)**

Mode:

The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Burst mode for the Regenerator Section (RS) error.

SINGLE, Single is selected as Burst mode.

REPEAT, Repeat is selected as Burst mode.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE
BERR

* SOUR:DATA:TEL:SDH:ERR:RS:BURS:MODE
SING

* SOUR:DATA:TEL:SDH:ERR:RS:BURS:MODE?
Returns SINGLE

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:MODE

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:DURation

Description

This command sets the duration for the Regenerator Section (RS) error.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:DURation <wsp> <Duration> | MAXimum
| MINimum

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:DURation

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the Regenerator Section (RS) error. Choices are 1 through 14400000.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:RS:BURS:MODE SING * SOUR:DATA:TEL:SDH:ERR:RS:BURS:DUR 15 * SOUR:DATA:TEL:SDH:ERR:RS:BURS:DUR? Returns 15</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:DURation?</pre>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:DURation?**

Description	<p>This query returns the duration for the Regenerator Section (RS) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:DURation?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<p><Duration></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:DURation?**

Response(s)	Duration: The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the duration for the Regenerator Section (RS) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:RS:BURS:MODE SING * SOUR:DATA:TEL:SDH:ERR:RS:BURS:DUR 15 * SOUR:DATA:TEL:SDH:ERR:RS:BURS:DUR? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:DURation

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:PERiod

Description

This command sets the period for the Regenerator Section (RS) error.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:PERiod <wsp> <Period> | MAXimum |
MINimum

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:PERiod****Parameter(s)**

Period:

The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum.

MAXimum allows to set the instrument to the greatest supported value.

MINimum allows to set the instrument to the smallest supported value.

Sets the period for the Regenerator Section (RS) error.

Choices are 1 through 14400000.

Example(s)

```
* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE  
BERR
```

```
* SOUR:DATA:TEL:SDH:ERR:RS:BURS:MODE  
REP
```

```
* SOUR:DATA:TEL:SDH:ERR:RS:BURS:PER 15
```

```
* SOUR:DATA:TEL:SDH:ERR:RS:BURS:PER?
```

```
Returns 15
```

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:PERiod**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:PERiod?**

Description	<p>This query returns the period for the Regenerator Section (RS) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:PERiod? [<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<p><Period></p>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:PERiod?

Response(s)

Period:

The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the period for the Regenerator Section (RS) error.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE

BERR

* SOUR:DATA:TEL:SDH:ERR:RS:BURS:MODE

REP

* SOUR:DATA:TEL:SDH:ERR:RS:BURS:PER 15

* SOUR:DATA:TEL:SDH:ERR:RS:BURS:PER?

Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt:PERiod

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt**

Description This command enables or disables the Burst for the Regenerator Section (RS) error.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt <wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the Regenerator Section (RS) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:RS:BURS ON* SOUR:DATA:TEL:SDH:ERR:RS:BURS? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS:
BURSt?**

Description	This query returns the status of Burst for the Regenerator Section (RS) error. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt?

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status for the Regenerator Section (RS) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:RS:BURS:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:RS:BURS ON * SOUR:DATA:TEL:SDH:ERR:RS:BURS? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:RS: BURSt

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:TYPE**

Description	This command selects the Burst type for the Regenerator Section (RS) alarm. At *RST, this value is set to LOF1.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE<wsp>LOF1 OOF2

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE

Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF1 OOF2.</p> <p>Selects the Burst type for the Regenerator Section (RS) error.</p> <p>LOF1, selects Loss of Frame (LOF1) as RS alarm.</p> <p>OOF2, selects Out of Frame (OOF1) as RS alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE LOF1</p> <p>* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE? Returns LOF1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:TYPE?**

Description	This query returns the Burst type for the Regenerator Section (RS) alarm. At *RST, this value is set to LOF1.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt?
Parameter(s)	None
Response Syntax	<Alarm>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:TYPE?**

Response(s)	Alarm: The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the Regenerator Section (RS) error. LOF1, Loss of Frame (LOF1) is selected as RS alarm. OOF2, Out of Frame (OOF1) is selected as RS alarm.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE LOF1 * SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE? Returns LOF1
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:MODE****Description**

This command selects the Burst mode for the Regenerator Section (RS) alarm.

At *RST, this value is set to SINGLE.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:MODE<wsp>SINGle|REPeat

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:MODE

Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat.</p> <p>Selects the Burst mode for the Regenerator Section (RS) alarm.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE LOF1 * SOUR:DATA:TEL:SDH:ALAR:RS:BURS:MODE SING * SOUR:DATA:TEL:SDH:ALAR:RS:BURS:MODE? Returns SINGLE</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:MODE?</pre>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:MODE?**

Description	This query returns the Burst mode for the Regenerator Section (RS) error. At *RST, this value is set to SINGLE.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:MODE?
Parameter(s)	None:
Response Syntax	<Mode>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:MODE?

Response(s)

Mode:

The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the Regenerator Section (RS) alarm.

SINGLE, Single is selected as Burst mode.

REPEAT, Repeat is selected as Burst mode.

Example(s)

```
* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE  
LOF1
```

```
* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:MODE  
SING
```

```
* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:MODE?  
Returns SINGLE
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:  
BURSt:TYPE
```

```
* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:  
BURSt:MODE
```

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:DURation****Description**

This command sets the duration for the Regenerator Section (RS) alarm.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:DURation <wsp> <Duration> | MAXimum
| MINimum

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:DURation**

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the Regenerator Section (RS) alarm.</p> <p>Choices are 1 through 14400000.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:DUR 15</p> <p>* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:DUR? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:DURation?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:DURation?**

Description	<p>This query returns the duration for the Regenerator Section (RS) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:DURation?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<pre><Duration></pre>

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:DURation?

Response(s)	Duration: The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the duration for the Regenerator Section (RS) alarm.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE BERR * SOUR:DATA:TEL:SDH:ALAR:RS:BURS:MODE SING * SOUR:DATA:TEL:SDH:ALAR:RS:BURS:DUR 15 * SOUR:DATA:TEL:SDH:ALAR:RS:BURS:DUR? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:DURation

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:PERiod****Description**

This command sets the period for the Regenerator Section (RS) alarm.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:PERiod <wsp> <Period> | MAXimum |
MINimum

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:PERiod

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the period for the Regenerator Section (RS) alarm.</p> <p>Choices are 1 through 14400000.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE BERR * SOUR:DATA:TEL:SDH:ALAR:RS:BURS:MODE REP * SOUR:DATA:TEL:SDH:ALAR:RS:BURS:PER 15 * SOUR:DATA:TEL:SDH:ALAR:RS:BURS:PER? Returns 15</pre>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:PERiod**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:PERiod?

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:PERiod?

Description This query returns the period for the Regenerator Section (RS) alarm.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:PERiod? [<wsp>MAXimum|MINimum]

Parameter(s) The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum|MINimum.
MAXimum is used to retrieve the instrument's greatest supported value.
MINimum is used to retrieve the instrument's smallest supported value.
This parameter is optional. If no token is specified, the current period will be returned.

Response Syntax <Period>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt:PERiod?****Response(s)**

Period:

The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the period for the Regenerator Section (RS) alarm.

Example(s)

```
* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE  
BERR
```

```
* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:MODE  
REP
```

```
* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:PER 15
```

```
* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:PER?
```

Returns 15

See Also

```
* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:  
BURSt:TYPE
```

```
* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:  
BURSt:MODE
```

```
* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:  
BURSt:PERiod
```

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt

Description This command enables or disables the Burst for the Regenerator Section (RS) alarm.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt<wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the Regenerator Section (RS) alarm.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE BERR* SOUR:DATA:TEL:SDH:ALAR:RS:BURS ON* SOUR:DATA:TEL:SDH:ALAR:RS:BURS Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt?

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt?

Description	This query returns the status of Burst for the Regenerator Section (RS) alarm. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS:
BURSt?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status for the Regenerator Section (RS) alarm.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:RS:BURS:TYPE BERR * SOUR:DATA:TEL:SDH:ALAR:RS:BURS ON * SOUR:DATA:TEL:SDH:ALAR:RS:BURS Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm:RS: BURSt

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:TYPE

Description	<p>This command selects the Burst type for the Multiplex Section (MS) error.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:TYPE<wsp>BERRor MSRei</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BERRor MSRei. Selects the Burst type for the Multiplex Section (MS) error.</p> <p>BERRor, selects B1 (BERROR) as MS error.</p> <p>MSRei, selects Multiplex Section - Remote Error Indicator (MSREI) as MS error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:TYPE?</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:TYPE?

Description	This query returns the Burst type for the Multiplex Section (MS) error. At *RST, this value is set to BERRor.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst type for the Multiplex Section (MS) error.</p> <p>BERRor, B1 (BERROR) is selected as MS error.</p> <p>MSRei, Multiplex Section - Remote Error Indicator (MSREI) is selected as MS error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE? Returns BERROR</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:TYPE</p>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:MODE

Description This command selects the Burst mode for the Multiplex Section (MS) error.

At *RST, this value is set to SINGLE.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:MODE<wsp>SINGle|REPeat

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:MODE**

Parameter(s)	Mode: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: Selects the Burst mode for the Multiplex Section (MS) error. SINGLE, selects Single as Burst mode. REPeat, selects Repeat as Burst mode.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:MS:BURS:MODE SING * SOUR:DATA:TEL:SDH:ERR:MS:BURS:MODE? Returns SINGLE
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:MODE?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:MODE?**

Description	This query returns the Burst mode for the Multiplex Section (MS) error. At *RST, this value is set to SINGLE.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:MODE?
Parameter(s)	None
Response Syntax	<Mode>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:MODE?****Response(s)**

Mode:

The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the Multiplex Section (MS) error.

SINGLE, Single is selected as Burst mode.

REPEAT, Repeat is selected as Burst mode.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE
BERR

* SOUR:DATA:TEL:SDH:ERR:MS:BURS:MODE
SING

* SOUR:DATA:TEL:SDH:ERR:MS:BURS:MODE?
Returns SINGLE

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:MODE

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:DURation

Description This command sets the duration for the Multiplex Section (MS) error.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:DURation <wsp> <Duration> | MAXimum
| MINimum

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:DURation**

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the duration for the Multiplex Section (MS) error.</p> <p>Choices are 1 through 14400000.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:DUR 15</p> <p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:DUR? Returns 15</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:DURation**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:DURation?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:DURation?**

Description	<p>This query returns the duration for the Multiplex Section (MS) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:DURation?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current duration will be returned.</p>
Response Syntax	<p><Duration></p>

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:DURation?

Response(s)

Duration:

The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the duration for the Multiplex Section (MS) error.

Example(s)

* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE

BERR

* SOUR:DATA:TEL:SDH:ERR:MS:BURS:MODE

SING

* SOUR:DATA:TEL:SDH:ERR:MS:BURS:DUR 15

* SOUR:DATA:TEL:SDH:ERR:MS:BURS:DUR?

Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:DURation

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:PERiod**

Description This command sets the period for the Multiplex Section (MS) error.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:PERiod <wsp> <Period> | MAXimum |
MINimum

:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:PERiod

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the Multiplex Section (MS) error.</p> <p>Choices are 1 through 14400000.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:MS:BURS:MODE REP* SOUR:DATA:TEL:SDH:ERR:MS:BURS:PER 15* SOUR:DATA:TEL:SDH:ERR:MS:BURS:PER? Returns 15

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:PERiod**

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:MODE

* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:PERiod?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:PERiod?**

Description	<p>This query returns the period for the Multiplex Section (MS) error.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:PERiod?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<p><Period></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt:PERiod?**

Response(s)	<p>Period:</p> <p>The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the period for the Multiplex Section (MS) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE BERR</p> <p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:MODE REP</p> <p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:PER 15</p> <p>* SOUR:DATA:TEL:SDH:ERR:MS:BURS:PER? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:PERiod</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt**

Description This command enables or disables the Burst for the Multiplex Section (MS) error.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt<wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the Multiplex Section (MS) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE BERR* SOUR:DATA:TEL:SDH:ERR:MS:BURS ON* SOUR:DATA:TEL:SDH:ERR:MS:BURS? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt:TYPE* SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt?

**:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS:
BURSt?**

Description	This query returns the status of Burst for the Multiplex Section (MS) error. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ERRor:MS: BURSt?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELecom:SDH:ERRor:MS:
BURSt?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status for the Multiplex Section (MS) error.
Example(s)	* SOUR:DATA:TEL:SDH:ERR:MS:BURS:TYPE BERR * SOUR:DATA:TEL:SDH:ERR:MS:BURS ON * SOUR:DATA:TEL:SDH:ERR:MS:BURS? Returns 1
See Also	* SOURce[1..n]:DATA:TELecom:SDH:ERRor:MS: BURSt:TYPE * SOURce[1..n]:DATA:TELecom:SDH:ERRor:MS: BURSt

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:TYPE

Description	<p>This command selects the Burst type for the Multiplex Section (MS) alarm.</p> <p>At *RST, this value is set to MSAis.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:TYPE<wsp>MSAis MSRDi</p>

**:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:TYPE**

Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MSAis MSRDi. Selects the Burst type for the Multiplex Section (MS) alarm.</p> <p>MSAis, selects Multiplex Section - Alarm Indication Signal (MSAIS) as Multiplex Section (MS) alarm.</p> <p>MSRDi, selects Multiplex Section - Remote Defect Indication (MSRDI) as Multiplex Section (MS) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE MSA</p> <p>* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE? Returns MSAIS</p>
See Also	<p>* SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:TYPE?**

Description	This query returns the Burst type for the Multiplex Section (MS) alarm. At *RST, this value is set to MSAis.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:TYPE?****Response(s)**

Alarm:

The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Selects the Burst type for the Multiplex Section (MS) alarm.

MSAis, Multiplex Section - Alarm Indication Signal (MSAIS) is selected as MS alarm.

MSRD_i, Multiplex Section - Remote Defect Indication (MSRDI) is selected as MS alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE
MSA

* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE?
Returns MSAIS

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
MS:BURSt:TYPE

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:MODE

Description

This command selects the mode for the Multiplex Section (MS) alarm.

At *RST, this value is set to SINGLE.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:MODE<wsp>SINGLE|REPeat

**:SOURCE[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:MODE**

Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGle REPeat.</p> <p>Selects the Burst mode for the Multiplex Section (MS) alarm.</p> <p>SINGle, selects Single as Burst mode.</p> <p>REPeat, selects Repeat as Burst mode.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE MSA</p> <p>* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:MODE? Returns SINGLE</p>
See Also	<p>* SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:TYPE</p> <p>* SOURCE[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:MODE?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:MODE?**

Description	This query returns the mode for the Multiplex Section (MS) alarm. At *RST, this value is set to SINGLE.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:MODE?
Parameter(s)	None
Response Syntax	<Mode>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:MODE?****Response(s)**

Mode:

The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Burst mode for the Multiplex Section (MS) alarm.

SINGLE, Single is selected as Burst mode.

REPEAT, Repeat is selected as Burst mode.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE
MSA

* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:MODE
SING

* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:MODE?
Returns SINGLE

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
MS:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
MS:BURSt:MODE

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:DURation

Description This command sets the duration for the Multiplex Section (MS) alarm.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:DURation<wsp><Duration> |MAXimum
|MINimum

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:DURation

Parameter(s)	<p>Duration:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the duration for the Multiplex Section (MS) alarm.</p> <p>Choices are 1 through 14400000.</p>
Example(s)	<p>* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE MSA</p> <p>* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:MODE SING</p> <p>* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:DUR 15</p> <p>* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:DUR? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:MODE</p> <p>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:DURation?</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:DURation?**

Description This query returns the duration for the Multiplex Section (MS) alarm.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:DURation?[<wsp>MAXimum | MINimum]

Parameter(s) The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum | MINimum.
MAXimum is used to retrieve the instrument's greatest supported value.
MINimum is used to retrieve the instrument's smallest supported value.
This parameter is optional. If no token is specified, the current duration will be returned.

Response Syntax <Duration>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:DURation?**

Response(s)	Duration: The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the duration for the Multiplex Section (MS) alarm.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE MSA * SOUR:DATA:TEL:SDH:ALAR:MS:BURS:MODE SING * SOUR:DATA:TEL:SDH:ALAR:MS:BURS:DUR 15 * SOUR:DATA:TEL:SDH:ALAR:MS:BURS:DUR? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:DURation

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:PERiod

Description This command sets the period for the Multiplex Section (MS) alarm.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:PERiod <wsp> <Period> |MAXimum
|MINimum

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:PERiod

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the period for the Multiplex Section (MS) alarm.</p> <p>Choices are 1 through 14400000.</p>
Example(s)	<pre>* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE MSA * SOUR:DATA:TEL:SDH:ALAR:MS:BURS:MODE REP * SOUR:DATA:TEL:SDH:ALAR:MS:BURS:PER 15 * SOUR:DATA:TEL:SDH:ALAR:MS:BURS:PER? Returns 15</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:PERiod?</pre>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:PERiod?**

Description	<p>This query returns the period for the Multiplex Section (MS) alarm.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt:DURation?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period will be returned.</p>
Response Syntax	<p><Period></p>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt:PERiod?**

Response(s)	Period: The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the period for the Multiplex Section (MS) alarm.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE MSA * SOUR:DATA:TEL:SDH:ALAR:MS:BURS:MODE REP * SOUR:DATA:TEL:SDH:ALAR:MS:BURS:PER 15 * SOUR:DATA:TEL:SDH:ALAR:MS:BURS:PER? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:MODE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:PERiod

:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt

Description This command enables or disables the Burst for the Multiplex Section (MS) alarm.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt<wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt**

Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Burst for the Multiplex Section (MS) alarm.
Example(s)	* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE MSA * SOUR:DATA:TEL:SDH:ALAR:MS:BURS ON * SOUR:DATA:TEL:SDH:ALAR:MS:BURS? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt:TYPE * SOURce[1..n]:DATA:TELEcom:SDH:ALARm: MS:BURSt?

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt?**

Description	This query returns the status of Burst for the Multiplex Section (MS) alarm. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS: BURSt?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:SDH:ALARm:MS:
BURSt?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status for the Multiplex Section (MS) alarm.

Example(s)

* SOUR:DATA:TEL:SDH:ALAR:MS:BURS:TYPE
MSA

* SOUR:DATA:TEL:SDH:ALAR:MS:BURS ON

* SOUR:DATA:TEL:SDH:ALAR:MS:BURS?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
MS:BURSt:TYPE

* SOURce[1..n]:DATA:TELEcom:SDH:ALARm:
MS:BURSt

:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]

Description	<p>This command sets the Regenerator Overhead byte values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n] <wsp> <Channel> ,A1 A2 J0 Z0 B1 E1 F1 D1 D2 D3 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33, <Value></pre>
Parameter(s)	<p>Channel:</p> <p>The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the regenerator overhead channel number.</p>

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]**

Overhead:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

A1|A2|J0|Z0|B1|E1|F1|D1|D2|D3|UD11|UD12
|UD13|UD21|UD22|UD23|UD31|UD32|UD33.

Selects the Regenerator Overhead bytes.

A1, selects F6 as a hexadecimal value for A1.

A2, selects 28 as a hexadecimal value for A2.

J0, selects the J0 trace.

Z0, selects Z0 as a Growth.

B1, selects B1 as a Bit Interleaved Parity code (BIP-8).

E1, selects E1 as an Orderwire.

F1, selects F1 as an User.

D1, selects D1 as a Data Communications Channel (DCC).

D2, selects D2 as a Data Communications Channel (DCC).

D3, selects D3 as a Data Communications Channel (DCC).

Byte is specified in two ways.

In first method standard names are used.

Ex: A1, A2.

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.

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]**

Value:

The program data syntax for the third parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Selects the regenerator overhead byte values in hexadecimal format.

Example(s)

* SOUR:DATA:TEL:SDH:OH:RS:OVER1 1,A1,#HF6
* SOUR:DATA:TEL:SDH:OH:RS:OVER1? 1,A1
Returns #HF6

Note

The suffix identifies a set of columns within the selected STM channel. A value of 1 selects columns 1,4 and 7, a value of 2 selects columns 2,5 and 8, and a value of 3 selects columns 3,6 and 9.

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]?

:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]?

Description	<p>This query returns the Regenerator Overhead byte values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]?<wsp><Channel>,A1 A2 J0 Z0 B1 E1 F1 D1 D2 D3 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33</pre>
Parameter(s)	<p>Channel:</p> <p>The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the regenerator overhead channel number.</p>

:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]?

Overhead:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

A1|A2|J0|Z0|B1|E1|F1|D1|D2|D3|UD11|UD12
|UD13|UD21|UD22|UD23|UD31|UD32|UD33.

Selects the regenerator overhead bytes.

A1, selects F6 as a hexadecimal value for A1.

A2, selects 28 as a hexadecimal value for A2.

J0, selects J0 trace.

Z0, selects Z0 as a Growth.

B1, selects B1 as a Bit Interleaved Parity code (BIP-8).

E1, selects E1 as an Orderwire.

F1, selects F1 as an User.

D1, selects D1 as a Data Communications Channel (DCC).

D2, selects D2 as a Data Communications Channel (DCC).

D3, selects D3 as a Data Communications Channel (DCC).

Byte is specified in two ways.

In first method standard names are used.

Ex: A1, A2.

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.

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]?****Response Syntax** <Value>**Response(s)** Value:
The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.
Returns the regenerator overhead byte value.**Example(s)** * SOUR:DATA:TEL:SDH:OH:RS:OVER1 1,A1,#HF6
* SOUR:DATA:TEL:SDH:OH:RS:OVER1? 1,A1
Returns #HF6**Note** The suffix identifies a set of columns within the selected STM channel. A value of 1 selects columns 1,4 and 7, a value of 2 selects columns 2,5 and 8, and a value of 3 selects columns 3,6 and 9.**See Also** * SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]

:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS: DISabled

Description	<p>This command clears an Overwrite for all bytes of Regenerator Overhead.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS: DISabled
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SDH:OH:RS:DIS

:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]:ENABLEd

Description	<p>This command enables or disables the generation of the selected regenerator byte.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]:ENABLEd<wsp><Channel>,<A1 A2 J0 Z0 B1 E1 F1 D1 D2 D3 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33, <Set></pre>
Parameter(s)	<p>Channel:</p> <p>The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the regenerator overhead channel number.</p> <p>Overhead:</p> <p>The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>A1 A2 J0 Z0 B1 E1 F1 D1 D2 D3 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33.</pre>

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]:ENABLEd**

Selects the regenerator overhead bytes.

A1, selects F6 as a hexadecimal value for A1.

A2, selects 28 as a hexadecimal value for A2.

J0, selects J0 trace.

Z0, selects Z0 as a Growth.

B1, selects B1 as a Bit Interleaved Parity code (BIP-8).

E1, selects E1 as an Orderwire.

F1, selects F1 as an User.

D1, selects D1 as a Data Communications Channel (DCC).

D2, selects D2 as a Data Communications Channel (DCC).

D3, selects D3 as a Data Communications Channel (DCC).

Byte is specified in two ways.

In first method standard names are used.

Ex: A1, A2.

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.

Set:

The program data syntax for the third parameter is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability.

ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the generation of the selected regenerator byte.

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]:ENABLEd****Example(s)**

* SOUR:DATA:TEL:SDH:OH:RS:OVER1:ENAB
1,A1,ON
* SOUR:DATA:TEL:SDH:OH:RS:OVER1: ENAB?
1,A1 Returns 1

Note

The suffix identifies a set of columns within the selected STM channel. A value of 1 selects columns 1,4 and 7, a value of 2 selects columns 2,5 and 8, and a value of 3 selects columns 3,6 and 9.

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]
* SOURce[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERwrite[1..n]:ENABLEd?

**:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]:ENABLEd?**

Description This query returns the status of generation of the selected regenerator byte.

At *RST, this value is set to OFF.

Syntax :SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]:ENABLEd?<wsp><Channel>,
A1|A2|J0|Z0|B1|E1|F1|D1|D2|D3|UD11|UD12
|UD13|UD21|UD22|UD23|UD31|UD32|UD33

Parameter(s) Channel:
The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.
Selects the regenerator overhead channel number.

Overhead:
The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
A1|A2|J0|Z0|B1|E1|F1|D1|D2|D3|UD11|UD12
|UD13|UD21|UD22|UD23|UD31|UD32|UD33.

**:SOURCE[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]:ENABLEd?**

Selects the regenerator overhead bytes.

A1, selects F6 as a hexadecimal value for A1.

A2, selects 28 as a hexadecimal value for A2.

J0, selects J0 trace.

Z0, selects Z0 as a Growth.

B1, selects B1 as a Bit Interleaved Parity code (BIP-8).

E1, selects E1 as an Orderwire.

F1, selects F1 as an User.

D1, selects D1 as a Data Communications Channel (DCC).

D2, selects D2 as a Data Communications Channel (DCC).

D3, selects D3 as a Data Communications Channel (DCC).

Byte is specified in two ways.

In first method standard names are used.

Ex: A1, A2.

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.

Response Syntax <Set>

:SOURce[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]:ENABLEd?

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of generation of the selected regenerator byte.
Example(s)	* SOUR:DATA:TEL:SDH:OH:RS:OVER1:ENAB 1,A1,ON * SOUR:DATA:TEL:SDH:OH:RS:OVER1:ENAB? 1,A1 Returns 1
Note	The suffix identifies a set of columns within the selected STM channel. A value of 1 selects columns 1,4 and 7, a value of 2 selects columns 2,5 and 8, and a value of 3 selects columns 3,6 and 9.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n] * SOURce[1..n]:DATA:TELEcom:SDH:OH:RS: OVERwrite[1..n]:ENABLEd

**:FETCh[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERwrite:STATus?**

Description	<p>This query returns the status of the regenerator byte in any timeslot having the overwrite selected or not.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDH:OH:RS: OVERwrite:STATus?
Parameter(s)	None
Response Syntax	<Status>
Response(s)	<p>Status:</p> <p>The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> element. Returns the status of the overwrite for regenerator overhead.</p> <p>PRESENT, indicates the presence of output signal at an optical port.</p> <p>ABSENT, indicates the absence of output signal at an optical port.</p>
Example(s)	<p>* FETC:DATA:TEL:SDH:OH:RS:OVER:STAT?</p> <p>Returns the status of overwrite for regenerator overhead.</p>

**:SENSe[1..n]:DATA:TELEcom:SDH:OH:RS:
OVERhead[1..n]?**

Description	<p>This query returns the Regenerator Overhead byte values in hexadecimal format for the receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDH:OH:RS: OVERhead[1..n]?<wsp> <Channel> ,A1 A2 J0 Z0 B1 E1 F1 D1 D2 D3 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33</pre>
Parameter(s)	<p>Channel: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the regenerator overhead channel number.</p> <p>Overhead: The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: A1 A2 J0 Z0 B1 E1 F1 D1 D2 D3 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33.</p>

**:SENSe[1..n]:DATA:TELecom:SDH:OH:RS:
OVERhead[1..n]?**

Selects the Regenerator Overhead bytes for the receiver.

A1, selects F6 as a hexadecimal value for A1.

A2, selects 28 as a hexadecimal value for A2.

J0, selects J0 trace.

Z0, selects Z0 as a Growth.

B1, selects B1 as a Bit Interleaved Parity code (BIP-8).

E1, selects E1 as an Orderwire.

F1, selects F1 as an User.

D1, selects D1 as a Data Communications Channel (DCC).

D2, selects D2 as a Data Communications Channel (DCC).

D3, selects D3 as a Data Communications Channel (DCC).

Byte is specified in two ways.

In first method standard names are used.

Ex: A1, A2.

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.

Response Syntax <Value>

:SENSe[1..n]:DATA:TELeom:SDH:OH:RS: OVERhead[1..n]?

Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the regenerator overhead byte value for the receiver.</p>
Example(s)	<p>* SENS:DATA:TEL:SDH:OH:RS:OVER1? 1,A1</p> <p>Returns the regenerator overhead byte value for the receiver.</p>
Note	<p>The suffix identifies a set of columns within the selected STM channel. A value of 1 selects columns 1,4 and 7, a value of 2 selects columns 2,5 and 8, and a value of 3 selects columns 3,6 and 9.</p>
See Also	<p>* SOURce[1..n]:DATA:TELeom:SDH:OH:RS:OVERhead[1..n]</p>

:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]

Description	<p>This command sets the Multiplexer Overhead byte values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead<wsp><Channel>,H1 H2 H3 B2 K1 K2 D4 D5 D6 D7 D8 D9 D10 D11 D12 S1 M0 M1 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63,<Value></pre>
Parameter(s)	<p>Channel:</p> <p>The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the multiplexer overhead channel number.</p> <p>Overhead:</p> <p>The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>H1 H2 H3 B2 K1 K2 D4 D5 D6 D7 D8 D9 D10 D11 D12 S1 M0 M1 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63.</pre>

:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]

Selects the multiplexer overhead bytes.

H1, selects H1 as a Pointer.

H2, selects H2 as a Pointer.

H3, selects H3 as a Pointer Action.

B2, selects B2 as a Bit Interleaved Parity code (BIP-8).

K1, selects K1 as an Automatic Protection Switching (APS).

K2, selects K2 as an Automatic Protection Switching (APS).

D4, selects D4 as a Data Communications Channel (DCC).

D5, selects D5 as a Data Communications Channel (DCC).

D6, selects D6 as a Data Communications Channel (DCC).

D7, selects D7 as a Data Communications Channel (DCC).

D8, selects D8 as a Data Communications Channel (DCC).

D9, selects D9 as a Data Communications Channel (DCC).

D10, selects D10 as a Data Communications Channel (DCC).

D11, selects D11 as a Data Communications Channel (DCC).

D12, selects D12 as a Data Communications Channel (DCC).

S1, selects S1 as a Synchronization status.

M0, selects M0 as a Remote Error Indicator - Line (REI-L).

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERhead[1..n]**

M1, selects M1 as a Remote Error Indicator - Line (REI-L).

E2, selects E2 as an Orderwire.

Byte is specified in two ways.

In first method standard names are used.

Ex: H1, H2.

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.

Value:

The program data syntax for the third parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Selects the multiplexer overhead byte values in hexadecimal format.

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERhead[1..n]**

Example(s) * SOUR:DATA:TEL:SDH:OH:MS:OVER1
1,H1,#H00
* SOUR:DATA:TEL:SDH:OH:MS:OVER1? 1,H1
Returns #H00

Note The suffix identifies a set of columns within the
selected STM channel. A value of 1 selects
columns 1,4 and 7, a value of 2 selects columns
2,5 and 8, and a value of 3 selects columns 3,6
and 9.

See Also * SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERhead[1..n]?

:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]?

Description	<p>This query returns the Multiplexer Overhead byte values in hexadecimal format.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead?<wsp> <Channel>,H1 H2 H3 B2 K1 K2 D4 D5 D6 D7 D8 D9 D10 D11 D12 S1 M0 M1 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63</pre>
Parameter(s)	<p>Channel:</p> <p>The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the multiplexer overhead channel number.</p> <p>Overhead:</p> <p>The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>H1 H2 H3 B2 K1 K2 D4 D5 D6 D7 D8 D9 D10 D11 D12 S1 M0 M1 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63.</pre>

:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]?

Selects the multiplexer overhead bytes.

H1, selects H1 as a Pointer.

H2, selects H2 as a Pointer.

H3, selects H3 as a Pointer Action.

B2, selects B2 as a Bit Interleaved Parity code (BIP-8).

K1, selects K1 as an Automatic Protection Switching (APS).

K2, selects K2 as an Automatic Protection Switching (APS).

D4, selects D4 as a Data Communications Channel (DCC).

D5, selects D5 as a Data Communications Channel (DCC).

D6, selects D6 as a Data Communications Channel (DCC).

D7, selects D7 as a Data Communications Channel (DCC).

D8, selects D8 as a Data Communications Channel (DCC).

D9, selects D9 as a Data Communications Channel (DCC).

D10, selects D10 as a Data Communications Channel (DCC).

D11, selects D11 as a Data Communications Channel (DCC).

D12, selects D12 as a Data Communications Channel (DCC).

S1, selects S1 as a Synchronization status.

M0, selects M0 as a Remote Error Indicator - Line (REI-L).

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERhead[1..n]?**

M1, selects M1 as a Remote Error Indicator - Line (REI-L).

E2, selects E2 as an Orderwire.

Byte is specified in two ways.

In first method standard names are used.

Ex: H1, H2.

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.

Response Syntax <Value>

Response(s) Value:
The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.
Returns the multiplexer overhead byte value.

:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:SDH:OH:MS:OVER1 1,H1,#H00* SOUR:DATA:TEL:SDH:OH:MS:OVER1? 1,H1 Returns #H00
Note	The suffix identifies a set of columns within the selected STM channel. A value of 1 selects columns 1,4 and 7, a value of 2 selects columns 2,5 and 8, and a value of 3 selects columns 3,6 and 9.
See Also	* SOURce[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
DISabled**

Description	<p>This command clears an Overwrite for all bytes of multiplexer overhead.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS: DISabled
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:SDH:OH:MS:DIS

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite[1..n]:ENABLEd**

Description	<p>This command enables or disables the generation of the selected multiplexer byte.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS: OVERwrite[1..n]:ENABLEd<wsp><Channel>, H1 H2 H3 B2 K1 K2 D4 D5 D6 D7 D8 D9 D10 D11 D12 S1 M0 M1 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63,<Set></pre>
Parameter(s)	<p>Channel: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the multiplexer overhead channel number.</p>

**:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite[1..n]:ENABLEd**

Overhead:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

H1|H2|H3|B2|K1|K2|D4|D5|D6|D7|D8|D9|
D10|D11|D12|S1|M0|M1|E2|UD11|UD12|UD13
|UD21|UD22|UD23|UD31|UD32|UD33|UD41|
UD42|UD43|UD51|UD52|UD53|UD61|UD62|
UD63.

Selects the multiplexer overhead bytes.

H1, selects H1 as a Pointer.

H2, selects H2 as a Pointer.

H3, selects H3 as a Pointer Action.

B2, selects B2 as a Bit Interleaved Parity code (BIP-8).

K1, selects K1 as an Automatic Protection Switching (APS).

K2, selects K2 as an Automatic Protection Switching (APS).

D4, selects D4 as a Data Communications Channel (DCC).

D5, selects D5 as a Data Communications Channel (DCC).

D6, selects D6 as a Data Communications Channel (DCC).

D7, selects D7 as a Data Communications Channel (DCC).

D8, selects D8 as a Data Communications Channel (DCC).

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite[1..n]:ENABLEd**

D9, selects D9 as a Data Communications Channel (DCC).

D10, selects D10 as a Data Communications Channel (DCC).

D11, selects D11 as a Data Communications Channel (DCC).

D12, selects D12 as a Data Communications Channel (DCC).

S1, selects S1 as a Synchronization status.

M0, selects M0 as a Remote Error Indicator - Line (REI-L).

M1, selects M1 as a Remote Error Indicator - Line (REI-L).

E2, selects E2 as an Orderwire.

Byte is specified in two ways.

In first method standard names are used.

Ex: H1, H2.

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.

**:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite[1..n]:ENABLEd****Set:**

The program data syntax for the third parameter is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the generation of the selected multiplexer byte.

Example(s)

* SOUR:DATA:TEL:SDH:OH:MS:OVER1:ENAB
1,H1,ON
* SOUR:DATA:TEL:SDH:OH:MS:OVER1:ENAB?
1,H1 Returns 1

Note

The suffix identifies a set of columns within the selected STM channel. A value of 1 selects columns 1,4 and 7, a value of 2 selects columns 2,5 and 8, and a value of 3 selects columns 3,6 and 9.

See Also

* SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERhead[1..n]
* SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite[1..n]:ENABLEd?

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite[1..n]:ENABLEd?**

Description This query returns the status of generation of the selected multiplexer byte.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite[1..n]:ENABLEd?<wsp><Channel>,
H1|H2|H3|B2|K1|K2|D4|D5|D6|D7|D8|D9|
D10|D11|D12|S1|M0|M1|E2|UD11|UD12|UD13
|UD21|UD22|UD23|UD31|UD32|UD33|UD41|
UD42|UD43|UD51|UD52|UD53|UD61|UD62|
UD63

Parameter(s) Channel:
The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.
Selects the multiplexer overhead channel number.

:SOURCE[1..n]:DATA:TELEcom:SDH:OH:MS: OVERwrite[1..n]:ENABLEd?

Overhead:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

H1|H2|H3|B2|K1|K2|D4|D5|D6|D7|D8|D9|
D10|D11|D12|S1|M0|M1|E2|UD11|UD12|UD13
|UD21|UD22|UD23|UD31|UD32|UD33|UD41|
UD42|UD43|UD51|UD52|UD53|UD61|UD62|
UD63.

Selects the multiplexer overhead bytes.

H1, selects H1 as a Pointer.

H2, selects H2 as a Pointer.

H3, selects H3 as a Pointer Action.

B2, selects B2 as a Bit Interleaved Parity code (BIP-8).

K1, selects K1 as an Automatic Protection Switching (APS).

K2, selects K2 as an Automatic Protection Switching (APS).

D4, selects D4 as a Data Communications Channel (DCC).

D5, selects D5 as a Data Communications Channel (DCC).

D6, selects D6 as a Data Communications Channel (DCC).

D7, selects D7 as a Data Communications Channel (DCC).

D8, selects D8 as a Data Communications Channel (DCC).

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite[1..n]:ENABLEd?**

D9, selects D9 as a Data Communications Channel (DCC).
D10, selects D10 as a Data Communications Channel (DCC).
D11, selects D11 as a Data Communications Channel (DCC).
D12, selects D12 as a Data Communications Channel (DCC).
S1, selects S1 as a Synchronization status.
M0, selects M0 as a Remote Error Indicator - Line (REI-L).
M1, selects M1 as a Remote Error Indicator - Line (REI-L).
E2, selects E2 as an Orderwire.
Byte is specified in two ways.
In first method standard names are used.
Ex: H1, H2.
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.

Response Syntax <Set>

Response(s) Set:
The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.
Returns the status of the generation of the selected multiplexer byte.

**:SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite[1..n]:ENABLEd?****Example(s)**

* SOUR:DATA:TEL:SDH:OH:MS:OVER1:ENAB
1,H1,ON
* SOUR:DATA:TEL:SDH:OH:MS:OVER1:ENAB?
1,H1 Returns 1

Note

The suffix identifies a set of columns within the selected STM channel. A value of 1 selects columns 1,4 and 7, a value of 2 selects columns 2,5 and 8, and a value of 3 selects columns 3,6 and 9.

See Also

* SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERhead[1..n]
* SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite[1..n]:ENABLEd

**:FETCh[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERwrite:STATus?**

Description	<p>This query returns the status of the multiplexer byte in any timeslot having the overwrite selected or not.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDH:OH:MS: OVERwrite:STATus?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Status></p>
Response(s)	<p>Status: The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> element. Returns the status of the overwrite for multiplexer overhead. PRESENT, indicates the presence of output signal at an optical port. ABSENT, indicates the absence of output signal at an optical port.</p>
Example(s)	<p>* FETC:DATA:TEL:SDH:OH:MS:OVER:STAT? Returns the status of overwrite for multiplexer overhead.</p>

:SENSe[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]?

Description	<p>This query returns the Multiplexer Overhead byte values in hexadecimal format for the receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead?<wsp><Channel>,H1 H2 H3 B2 K1 K2 D4 D5 D6 D7 D8 D9 D10 D11 D12 S1 M0 M1 E2 UD11 UD12 UD13 UD21 UD22 UD23 UD31 UD32 UD33 UD41 UD42 UD43 UD51 UD52 UD53 UD61 UD62 UD63</pre>
Parameter(s)	<p>Channel:</p> <p>The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the multiplexer overhead channel number.</p>

**:SENSe[1..n]:DATA:TELEcom:SDH:OH:MS:
OVERhead[1..n]?**

Overhead:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

H1|H2|H3|B2|K1|K2|D4|D5|D6|D7|D8|D9|
D10|D11|D12|S1|M0|M1|E2|UD11|UD12|UD13
|UD21|UD22|UD23|UD31|UD32|UD33|UD41|
UD42|UD43|UD51|UD52|UD53|UD61|UD62|
UD63.

Selects the multiplexer overhead bytes.

H1, selects H1 as a Pointer.

H2, selects H2 as a Pointer.

H3, selects H3 as a Pointer Action.

B2, selects B2 as a Bit Interleaved Parity code (BIP-8).

K1, selects K1 as an Automatic Protection Switching (APS).

K2, selects K2 as an Automatic Protection Switching (APS).

D4, selects D4 as a Data Communications Channel (DCC).

D5, selects D5 as a Data Communications Channel (DCC).

D6, selects D6 as a Data Communications Channel (DCC).

D7, selects D7 as a Data Communications Channel (DCC).

D8, selects D8 as a Data Communications Channel (DCC).

:SENSe[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]?

D9, selects D9 as a Data Communications Channel (DCC).

D10, selects D10 as a Data Communications Channel (DCC).

D11, selects D11 as a Data Communications Channel (DCC).

D12, selects D12 as a Data Communications Channel (DCC).

S1, selects S1 as a Synchronization status.

M0, selects M0 as a Remote Error Indicator - Line (REI-L).

M1, selects M1 as a Remote Error Indicator - Line (REI-L).

E2, selects E2 as an Orderwire.

In first method standard names are used.

Ex: H1, H2.

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.

Response Syntax <Value>

Response(s) Value:
The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.
Returns the multiplexer overhead byte value for the receiver.

:SENSe[1..n]:DATA:TELEcom:SDH:OH:MS: OVERhead[1..n]?

Example(s) * SENS:DATA:TEL:SDH:OH:MS:OVER1? 1,H1
Returns the multiplexer overhead byte value for the receiver.

Note The suffix identifies a set of columns within the selected STM channel. A value of 1 selects columns 1,4 and 7, a value of 2 selects columns 2,5 and 8, and a value of 3 selects columns 3,6 and 9.

See Also * SOURce[1..n]:DATA:TELEcom:SDH:OH:MS:OVERhead[1..n]?

**:FETCh[1..n]:DATA:TELEcom:SDH:RS:PM:
STATistics?**

Description	<p>This query returns the performance monitoring statistics of Regenerator Section (RS).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SDH:RS:PM: STATistics?<wsp>G829ISM,EFS EB ES SES BBE UAS ESR SESr BBER,NEND</pre>
Parameter(s)	<p>Standard:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: G829ISM.</p> <p>Selects the performance monitoring standard number.</p> <p>G829ISM, selects G.829 ISM as a standard number.</p>

**:FETCh[1..n]:DATA:TELEcom:SDH:RS:PM:
STATistics?**

Type:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> element for this parameter is:

EFS|EB|ES|SES|BBE|UAS|ESR|SESR|BBER.

Selects the performance monitoring statistics.

EFS, selects Error Free Seconds (EFS).

EB, selects Errored Block (EB).

ES, selects Errored Seconds (ES).

SES, selects Severely Errored Seconds (SES).

BBE, selects Background Block Error (SES).

UAS, selects Unavailable Second (UAS).

ESR, selects Errored Second Ratio (ESR).

SESR, selects Severely Errored Second Ratio (SESR).

BBER, selects Background Block Error Ratio (BBER).

**:FETCh[1..n]:DATA:TELEcom:SDH:RS:PM:
STATistics?**

End:

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> element for this parameter is:NEND.

Selects the Near-End.

NEND, selects the standard for Near-End.

Response Syntax

<Statistics>

Response(s)

Statistics:

The response data syntax for <Statistics> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the performance monitoring statistics of Regenerator Section (RS).

Example(s)

* FETC:DATA:TEL:SDH:RS:PM:STAT?
G826ISM,EFS,NEND

Returns the performance monitoring statistics of RS.

**:FETCh[1..n]:DATA:TELEcom:SDH:MS:PM:
STATistics?**

Description This query returns the performance monitoring statistics of Multiplex Section (MS).

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:SDH:MS:PM:
STATistics?<wsp>G829ISM|M2101ISM,EFS|EB|
ES|SES|BBE|UAS|ESR|SESR|BBER,NEND|
FEND

Parameter(s) Standard:
The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
G829ISM|M2101ISM.
Selects the performance monitoring standard number.
G829ISM, selects G.829 ISM as a standard number.
M2101ISM, selects M.2101 ISM as a standard number.

**:FETCh[1..n]:DATA:TELecom:SDH:MS:PM:
STATistics?**

Type:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

EFS|EB|ES|SES|BBE|UAS|ESR|SESR|BBER.

Selects the performance monitoring statistics.

EFS, selects Error Free Seconds (EFS).

EB, selects Errored Block (EB).

ES, selects Errored Seconds (ES).

SES, selects Severely Errored Seconds (SES).

BBE, selects Background Block Error (SES).

UAS, selects Unavailable Second (UAS).

ESR, selects Errored Second Ratio (ESR).

SESR, selects Severely Errored Second Ratio (SESR).

BBER, selects Background Block Error Ratio (BBER).

**:FETCh[1..n]:DATA:TELEcom:SDH:MS:PM:
STATistics?**

End:

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

NEND|FEND.

Selects Near-End or Far-End.

NEND, selects the standard for Near-End.

FEND, selects the standard for Far-End.

Response Syntax <Statistics>

Response(s) Statistics:

The response data syntax for <Statistics> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the performance monitoring statistics of Multiplex Section (MS).

Example(s) * FETC:DATA:TEL:SDH:MS:PM:STAT?
G829ISM, EFS, NEND

Returns the performance monitoring statistics of MS.

OTN Command Reference

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: FEC

Description	<p>This command enables or disables the Forward Error Correction (FEC) for the Transmitter (TX) mode.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: FEC<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the FEC (Forward Error Correction) for the TX (Transmitter) mode.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
FEC**

Example(s) * SOUR:DATA:TEL:OTN:OTU1:FEC ON
 * SOUR:DATA:TEL:OTN:OTU1:FEC? Returns 1

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
 FEC?

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

Description	This query returns the status of Forward Error Correction (FEC) for the Transmitter (TX) mode. At *RST, this value is set to ON.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: FEC?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of Forward Error Correction (FEC).

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

Example(s) * SOUR:DATA:TEL:OTN:OTU1:FEC ON
 * SOUR:DATA:TEL:OTN:OTU1:FEC? Returns 1

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
 FEC

**:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
FEC**

Description	<p>This command enables or disables the Forward Error Correction (FEC) for the Receiver (RX) mode.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: FEC<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the FEC (Forward Error Correction) for the RX (Receiver) mode.</p>

**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:
FEC**

Example(s) * SENS:DATA:TEL:OTN:OTU1:FEC ON
 * SENS:DATA:TEL:OTN:OTU1:FEC? Returns 1

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:
 FEC?

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

Description	This query returns the status of Forward Error Correction (FEC) for the Receiver (RX) mode. At *RST, this value is set to ON.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: FEC?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of Forward Error Correction (FEC).

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

Example(s) * SENS:DATA:TEL:OTN:OTU1:FEC ON
 * SENS:DATA:TEL:OTN:OTU1:FEC? Returns 1

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
 FEC

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

Description	<p>This command enables or disables the scrambler for the Transmitter (TX) mode.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SCRambler<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the scrambler.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:SCR ON * SOUR:DATA:TEL:OTN:OTU1:SCR? Returns 1</pre>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SCRambler?</pre>

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

Description	This query returns the status of scrambler for the Transmitter (TX) mode. At *RST, this value is set to ON.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SCRambler?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of scrambler.

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
SCRambler?****Example(s)**

- * SOUR:DATA:TEL:OTN:OTU1:SCR ON
- * SOUR:DATA:TEL:OTN:OTU1:SCR? Returns 1

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

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

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

Description	<p>This command enables or disables the scrambler for the Receiver (RX) mode.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SCRambler<wsp><Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the scrambler.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OTU1:SCR ON * SENS:DATA:TEL:OTN:OTU1:SCR? Returns 1</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:SCRambler?</p>

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

Description	This query returns the status of scrambler for the Receiver (RX) mode. At *RST, this value is set to ON.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SCRambler?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of scrambler.

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

Example(s) * SENS:DATA:TEL:OTN:OTU1:SCR ON
 * SENS:DATA:TEL:OTN:OTU1:SCR? Returns 1

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
 SCRambler

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

Description	<p>This command allows editing the Source Access Point Identifier to be generated (TTI bytes 1 to 15) for the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:SAPI:B<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:TCM1:SAPI:B1 "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:ODU1:TCM1:SAPI:B1? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:SAPI:B?</pre>

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

Description	<p>This query returns the generated Source Access Point Identifier for the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:SAPI:B?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the generated Source Access Point Identifier.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:TCM1:SAPI:B1 "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:ODU1:TCM1:SAPI:B1? Returns "EXFO OTU SAPI"</p>
See Also	<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]:DAPI:B

Description	<p>This command allows editing the Destination Access Point Identifier to be generated (TTI bytes 17 to 31) for the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi DAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B<wsp><Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:TCM1:DAPI:B "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:ODU1:TCM1:DAPI:B? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM1[1..n]:DAPI:B?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B?

Description	<p>This query returns the generated Destination Access Point Identifier for the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM[1..n]:DAPI:B?</pre>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Destination Access Point Identifier.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:TCM1:DAPI:B "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:ODU1:TCM1:DAPI:B? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TCM1[1..n]:DAPI:B</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TCM[1..n]:OPSPec:B**

Description	<p>This command allows editing the Operator Specific to be generated (TTI bytes 32 to 63) for the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:OPSPec:B<wsp><Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:TCM1:OPSP:B "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:ODU1:TCM1:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:OPSPec:B?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TCM[1..n]:OPSPec:B?**

Description	<p>This query returns the generated Operator Specific for the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:OPSPec:B?
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Operator Specific.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:TCM1:OPSP:B "EXFO OTU OPERATOR SPECIFIC"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:TCM1:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</p>
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:OPSPec:B

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

Description	<p>This command allows editing the Source Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1e/2e of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E :TCM[1..n]:SAPI:B<wsp><Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:E:TCM1:SAPI:B1 "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:ODU1:E:TCM1:SAPI:B1? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:SAPI:B?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:TCM[1..n]:SAPI:B?**

Description	<p>This query returns the generated Source Access Point Identifier for ODU1e/2e of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:TCM[1..n]:SAPI:B?<wsp><Message></p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the generated Source Access Point Identifier for ODU1e/2e.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:E:TCM1:SAPI:B1 "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:ODU1:E:TCM1:SAPI:B1? Returns "EXFO OTU SAPI"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:TCM[1..n]:SAPI:B</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:TCM[1..n]:DAPI:B**

Description	<p>This command allows editing the Destination Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1e/2e of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:DAPI:B <wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:E:TCM1:DAPI:B "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:ODU1:E:TCM1:DAPI:B? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM1[1..n]:DAPI:B?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:DAPI:B?

Description	This query returns the generated Destination Access Point Identifier for ODU1e/2e of the TCM level. At *RST, this value is set to EXFO TCMi SAPI.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:TCM[1..n]:DAPI:B?<wsp><Message>
Parameter(s)	None
Response Syntax	<Message>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:TCM[1..n]:DAPI:B?**

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the generated Destination Access Point Identifier for ODU1e/2e.
Example(s)	* SOUR:DATA:TEL:OTN:ODU1:E:TCM1:DAPI:B "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:ODU1:E:TCM1:DAPI:B? Returns "EXFO OTU DAPI"
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM1[1..n]:DAPI:B

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:OPSPec:B

Description	<p>This command allows editing the Operator Specific to be generated (TTI bytes 32 to 63) for ODU1e/2e of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:OPSPec:B<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:E:TCM1:OPSP:B "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:ODU1:E:TCM1:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:OPSPec:B?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:TCM[1..n]:OPSPec:B?**

Description	<p>This query returns the generated Operator Specific for ODU1e/2e of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:TCM[1..n]:OPSPec:B? <wsp> <Message>
Parameter(s)	None
Response Syntax	<Message>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:OPSPec:B?

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the generated Destination Access Point Identifier for ODU1e/2e.
Example(s)	* SOUR:DATA:TEL:OTN:ODU1:E:TCM1:OPSP:B "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:ODU1:E:TCM1:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:OPSPec:B

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

Description	<p>This command sets the injected message of Source Access Point Identifier (SAPI) for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU SAPI"</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:SAPI:B16<wsp><Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the injected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:SM:SAPI:B16 "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:OTU1:SM:SAPI:B16? Returns "EXFO OTU SAPI"</pre>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:SAPI:B16?</pre>

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

Description	<p>This query returns the injected message of Source Access Point Identifier (SAPI) for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU SAPI"</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:SAPI:B16?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the injected message for the instrument.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
SM:SAPI:B16?****Example(s)**

* SOUR:DATA:TEL:OTN:OTU1:SM:SAPI:B16
"EXFO OTU SAPI"
* SOUR:DATA:TEL:OTN:OTU1:SM:SAPI:B16?
Returns "EXFO OTU SAPI"

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

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

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

Description	<p>This command sets the injected message of Destination Access Point Identifier (DAPI) for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:DAPI:B16<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the injected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:SM:DAPI:B16 "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:OTU1:SM:DAPI:B16? Returns "EXFO OTU DAPI"</pre>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:DAPI:B16?</pre>

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

Description	<p>This query returns the injected message of Destination Access Point Identifier (DAPI) for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:DAPI:B16?
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the injected message for the instrument.</p>

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

Example(s) * SOUR:DATA:TEL:OTN:OTU1:SM:DAPI:B16
 "EXFO OTU DAPI"
 * SOUR:DATA:TEL:OTN:OTU1:SM:DAPI:B16?
 Returns "EXFO OTU DAPI"

Note OTU1/OTU2 are available for FTB/QS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/QS-8140 Transport Blazer
 only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
 SM:DAPI:B16

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

Description	<p>This command sets the injected message for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OPSPec:B32 <wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:SM:OPSP:B32 "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:OTU1:SM:OPSP:B32? Returns "EXFO OTU OPERATOR SPECIFIC"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OPSPec:B32?</pre>

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

Description	<p>This query returns the injected message for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OPSPec:B32? <wsp></p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the selected message for the instrument.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OTU1:SM:OPSP:B32 "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:OTU1:SM:OPSP:B32? Returns "EXFO OTU OPERATOR SPECIFIC"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OPSPec:B32</p>

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

Description	<p>This command allows editing the Source Access Point Identifier to be generated (TTI bytes 1 to 15) for Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:SAPI:B<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:PM:SAPI:B "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:ODU1:PM:SAPI:B? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:SAPI:B?</pre>

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

Description	<p>This query returns the injected message of Operator Specific for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:SAPI:B?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the generated Source Access Point Identifier for Performance Monitoring.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:PM:SAPI:B "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:ODU1:PM:SAPI:B? Returns "EXFO OTU SAPI"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:SAPI:B</p>

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

Description	<p>This command allows editing the Destination Access Point Identifier to be generated (TTI bytes 17 to 31) for Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi DAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:DAPI:B<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:PM:DAPI:B "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:ODU1:PM:DAPI:B? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:DAPI:B?</pre>

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

Description	<p>This query returns the generated Destination Access Point Identifier for Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi DAPI.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:DAPI:B?</code>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Source Access Point Identifier for Performance Monitoring.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:PM:DAPI:B "EXFO OTU DAPI"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:PM:DAPI:B? Returns "EXFO OTU DAPI"</p>
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:DAPI:B

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

Description	<p>This command allows editing the Operator Specific to be generated (TTI bytes 32 to 63) for Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:OPSPec:B<wsp><Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:PM:OPSP:B "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:ODU1:PM:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:OPSPec:B?</pre>

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

Description	<p>This query returns the generated Operator Specific for Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:OPSPec:B?
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Operator Specific for Performance Monitoring.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:PM:OPSP:B "EXFO OTU OPERATOR SPECIFIC"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:PM:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</p>
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: PM:OPSPec:B

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:PM:SAPI:B

Description	<p>This command allows editing the Source Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1e/2e Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:PM:SAPI:B<wsp><Message>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the message for the instrument.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:E:PM:SAPI:B "EXFO OTU SAPI"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:E:PM:SAPI:B? Returns "EXFO OTU SAPI"</p>
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:PM:SAPI:B?

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

Description	This query returns the generated Source Access Point Identifier for ODU1e/2e Performance Monitoring. At *RST, this value is set to EXFO TCMi SAPI.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:PM:SAPI:B?
Parameter(s)	None
Response Syntax	<Message>

**:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:E
:PM:SAPI:B?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the generated Source Access Point Identifier for ODU1e/2e for Performance Monitoring.

Example(s)

* SOUR:DATA:TEL:OTN:ODU1:E:PM:SAPI:B
"EXFO OTU SAPI"

* SOUR:DATA:TEL:OTN:ODU1:E:PM:SAPI:B?
Returns "EXFO OTU SAPI"

See Also

* SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:
E:PM:SAPI:B

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

Description	<p>This command allows editing the Destination Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1e/2e Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: :PM:DAPI:B <wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:E:PM:DAPI:B "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:ODU1:E:PM:DAPI:B? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:PM:DAPI:B?</pre>

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

Description	This query returns the generated Destination Access Point Identifier for ODU1e/2e Performance Monitoring. At *RST, this value is set to EXFO TCMi SAPI.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:PM:DAPI:B?
Parameter(s)	None
Response Syntax	<Message>

:SOURce[1..n]:DATA:TELeom:OTN:ODU[1..n]:E :PM:DAPI:B?

Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Destination Access Point Identifier for ODU1e/2e for Performance Monitoring.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:E:PM:DAPI:B "EXFO OTU DAPI"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:E:PM:DAPI:B? Returns "EXFO OTU DAPI"</p>
See Also	<p>* SOURce[1..n]:DATA:TELeom:OTN:ODU[1..n]: E:PM:DAPI:B</p>

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

Description	<p>This command allows editing the Operator Specific to be generated (TTI bytes 32 to 63) for ODU1e/2e Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: :PM:OPSPec:B<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:E:PM:OPSP:B "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:ODU1:E:PM:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:PM:OPSPec:B?</pre>

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

Description	<p>This query returns the generated Destination Access Point Identifier for ODU1e/2e Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:PM:OPSPec:B?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the generated Operator Specific for ODU1e/2e for Performance Monitoring.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:E:PM:OPSP:B "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:ODU1:E:PM:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:PM:OPSPec:B</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
SM:OVERwrite:ENABLEd**

Description	<p>This command enables or disables the SM Overwrite feature for standard rates OTU1/2.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OVERwrite:ENABLEd <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the SM Overwrite feature.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:SM:OVER:ENAB ON * SOUR:DATA:TEL:OTN:OTU1:SM:OVER:ENAB? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OVERwrite:ENABLEd?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
SM:OVERwrite:ENABLEd?**

Description	<p>This query returns the status of SM Overwrite feature for standard rates OTU1/2.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OVERwrite:ENABLEd?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the SM Overwrite feature.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OTU1:SM:OVER:ENAB ON</p> <p>* SOUR:DATA:TEL:OTN:OTU1:SM:OVER:ENAB? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: SM:OVERwrite:ENABLEd</p>

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

Description	<p>This command sets the expected message of Source Access Point Identifier (SAPI) for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:SAPI:EXPEcted <wsp> <Message></pre>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the expected message for the instrument.</p>

:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]: TTI:SAPI:EXPEcted

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:OTN:OTU1:TTI:TIM SAPI,ON* SENS:DATA:TEL:OTN:OTU1:TTI:SAPI:EXP "EXFO OTU SAPI"* SENS:DATA:TEL:OTN:OTU1:TTI:SAPI:EXP? Returns "EXFO OTU SAPI"
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]: TTI:TIM* SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]: TTI:SAPI:EXPEcted?

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

Description	<p>This query returns the expected message of Source Access Point Identifier (SAPI) for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	<code>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:SAPI:EXPeCted?</code>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the expected message for the instrument.</p>

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

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:OTN:OTU1:TTI:TIM SAPI,ON* SENS:DATA:TEL:OTN:OTU1:TTI:SAPI:EXP "EXFO OTU SAPI"* SENS:DATA:TEL:OTN:OTU1:TTI:SAPI:EXP? Returns "EXFO OTU SAPI"
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]: TTI:TIM* SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]: TTI:SAPI:EXPeCted

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

Description	<p>This command sets the expected message of Destination Access Point Identifier (DAPI) for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:DAPI:EXPEcted <wsp> <Message ></pre>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the expected message for the instrument.</p>

:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]: TTI:DAPI:EXPEcted

Example(s) * SENS:DATA:TEL:OTN:OTU1:TTI:TIM DAPI,ON
 * SENS:DATA:TEL:OTN:OTU1:TTI:DAPI:EXP
 "EXFO OTU DAPI"
 * SENS:DATA:TEL:OTN:OTU1:TTI:DAPI:EXP?
 Returns "EXFO OTU DAPI"

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SENSE[1..n]:DATA:TELecom:OTN:OTU[1..n]:
 TTI:TIM
 * SENSE[1..n]:DATA:TELecom:OTN:OTU[1..n]:
 TTI:DAPI:EXPEcted?

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

Description	<p>This query returns the expected message of Destination Access Point Identifier (DAPI) for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:DAPI:EXPeCted?</pre>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the expected message for the instrument.</p>

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

Example(s) * SENS:DATA:TEL:OTN:OTU1:TTI:TIM DAPI,ON
 * SENS:DATA:TEL:OTN:OTU1:TTI:DAPI:EXP
 "EXFO OTU DAPI"
 * SENS:DATA:TEL:OTN:OTU1:TTI:DAPI:EXP?
 Returns "EXFO OTU DAPI"

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SENSE[1..n]:DATA:TELecom:OTN:OTU[1..n]:
 TTI:TIM
 * 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:SAPI:EXPeCted

Description	<p>This command allows editing the expected Source Access Point Identifier to be generated (TTI bytes 1 to 15) for the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: TCM[1..n]:TTI:SAPI:EXPeCted <wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:SAPI: EXP "EXFO OTU SAPI" * SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:SAPI: EXP? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: TCM[1..n]:TTI:TIM * SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]: TCM[1..n]:TTI:SAPI:EXPeCted?</pre>

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

Description	This query returns the expected Source Access Point Identifier for the TCM level. At *RST, this value is set to EXFO TCMi SAPI.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:SAPI:EXPeCted? <wsp> <Message>
Parameter(s)	None
Response Syntax	<Message>

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:
TCM[1..n]:TTI:SAPI:EXPeCted?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the expected Source Access Point Identifier.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:TIM
SAPI,ON

* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:SAPI:
EXP "EXFO OTU SAPI"

* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:SAPI:
EXP? Returns "EXFO OTU SAPI"

See Also

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

* SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]:
TCM[1..n]:TTI:SAPI:EXPeCted

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:DAPI:EXpected

Description	<p>This command allows editing the expected Destination Access Point Identifier to be generated (TTI bytes 17 to 31) for the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi DAPI.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:DAPI:EXpected<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:DAPI: EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:DAPI: EXP? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:TIM * SENSe[1..n]:DATA:TELEcom:OTN:ODU1[1..n]: TCM[1..n]:TTI:DAPI:EXpected?</pre>

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

Description	This query returns the Destination Access Point Identifier for the TCM level. At *RST, this value is set to EXFO TCMi DAPI.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:DAPI:EXPeCted? <wsp> <Message>
Parameter(s)	None
Response Syntax	<Message>

:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TCM[1..n]:TTI:DAPI:EXPeCted?

Response(s) Message:
The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.
Returns the expected Destination Access Point Identifier.

Example(s) * SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:TIM
DAPI,ON
* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:DAPI:
EXP "EXFO OTU DAPI"
* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:DAPI:
EXP? Returns "EXFO OTU DAPI"

See Also * SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:
TCM[1..n]:TTI:TIM
* SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]:
TCM[1..n]:TTI:DAPI:EXPeCted

:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:SAPI:EXPeCted

Description	<p>This command allows editing the expected Source Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1e/2e of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:SAPI:EXPeCted <wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the expected message for the instrument.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:SAPI: EXP "EXFO OTU SAPI" * SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:SAPI: EXP? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:TIM * SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]: E:TCM[1..n]:TTI:SAPI:EXPeCted?]</pre>

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

Description	This query returns the expected Source Access Point Identifier for ODU1e/2e of the TCM level. At *RST, this value is set to EXFO TCMi SAPI.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:SAPI:EXPeCted? <wsp> <Message>
Parameter(s)	None
Response Syntax	<Message>

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E:
TCM[1..n]:TTI:SAPI:EXPEcted?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the expected TCM level for non standard rates ODU1e/2e.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:TIM
SAPI,ON

* SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:SAPI:
EXP "EXFO OTU SAPI"

* SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:SAPI:
EXP? Returns "EXFO OTU SAPI"

See Also

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

* SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]:
E:TCM[1..n]:TTI:SAPI:EXPEcted

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

Description

This command allows editing the expected Destination Access Point Identifier to be generated (TTI bytes 17 to 31) for ODU1e/2e of the TCM level.

At *RST, this value is set to EXFO TCMi DAPI.

Syntax

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:
TCM[1..n]:TTI:DAPI:EXpected<wsp>
<Message>

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

Parameter(s)	Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the expected message for the instrument.
Example(s)	* SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:DAPI: EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:DAPI: EXP? Returns "EXFO OTU DAPI"
See Also	* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:TIM * SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]: E:TCM[1..n]:TTI:DAPI:EXPeCted?

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

Description	This query returns the selected TCM level for non standard rates ODU1e/2e. At *RST, this value is set to EXFO TCMi DAPI.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:DAPI:EXpected? <wsp> <Message>
Parameter(s)	None
Response Syntax	<Message>

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

Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the expected TCM level for non standard rates ODU1e/2e.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:TIM DAPI,ON</p> <p>* SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:DAPI: EXP "EXFO OTU DAPI"</p> <p>* SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:DAPI: EXP? Returns "EXFO OTU DAPI"</p>
See Also	<p>* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:TIM</p> <p>* SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]: E:TCM[1..n]:TTI:DAPI:EXPeCted</p>

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

Description This command allows editing the expected Source Access Point Identifier to be generated (TTI bytes 1 to 15).

At *RST, this value is set to EXFO ODU SAPI.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI
:SAPI:EXPEcted <wsp> <Message>

Parameter(s) Message:
The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.
Sets the expected message for the instrument.

Example(s) * SENS:DATA:TEL:OTN:ODU1:TTI:TIM SAPI,ON
* SENS:DATA:TEL:OTN:ODU1:TTI:SAPI:EXP
"EXFO OTU SAPI"
* SENS:DATA:TEL:OTN:ODU1:TTI:SAPI:EXP?
Returns "EXFO OTU SAPI"

See Also * SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TTI:TIM
* SENSE[1..n]:DATA:TELEcom:OTN:ODU1[1..n]:
TTI:SAPI:EXPEcted?

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

Description	This query returns the expected Source Access Point Identifier. At *RST, this value is set to EXFO ODU SAPI.
Syntax	:SENSe[1..n]:DAT:TELEcom:OTN:ODU[1..n]:TTI: SAPI:EXPEcted?<wsp><Message>
Parameter(s)	None
Response Syntax	<Message>

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

Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the expected Source Access Point Identifier.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:TTI:TIM SAPI,ON</p> <p>* SENS:DATA:TEL:OTN:ODU1:TTI:SAPI:EXP "EXFO OTU SAPI"</p> <p>* SENS:DATA:TEL:OTN:ODU1:TTI:SAPI:EXP?</p> <p>Returns "EXFO OTU SAPI"</p>
See Also	<p>* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:TIM</p> <p>* SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]:TTI:SAPI:EXPeCted</p>

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

Description	<p>This command allows editing the expected Destination Access Point Identifier to be generated (TTI bytes 1 to 15).</p> <p>At *RST, this value is set to EXFO ODU DAPI.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI :DAPI:EXPeCted <wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the expected message for the instrument.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:ODU1:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:ODU1:TTI:DAPI:EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:ODU1:TTI:DAPI:EXP? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TTI:TIM * SENSe[1..n]:DATA:TELEcom:OTN:ODU1[1..n]: TTI:DAPI:EXPeCted?</pre>

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

Description	This query returns the expected Destination Access Point Identifier. At *RST, this value is set to EXFO ODU DAPI.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI DAPI:EXPEcted?<wsp><Message>
Parameter(s)	None
Response Syntax	<Message>

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

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the expected Destination Access Point Identifier.
Example(s)	* SENS:DATA:TEL:OTN:ODU1:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:ODU1:TTI:DAPI:EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:ODU1:TTI:DAPI:EXP? Returns "EXFO OTU DAPI"
See Also	* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:TTI:TIM * SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]:TTI:DAPI:EXPeCted

:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI :SAPI:B?

Description	<p>This query returns the generated Source Access Point Identifier.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI :SAPI:B?<wsp><Message></pre>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Source Access Point Identifier.</p>
Example(s)	<pre>* FETC:DATA:TEL:OTN:ODU1:TTI:SAPI:B? Returns "EXFO OTU SAPI"</pre>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI
:DAPI:B?**

Description	<p>This query returns the generated Destination Access Point Identifier.</p> <p>At *RST, this value is set to EXFO TCMi DAPI.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI :DAPI:B? <wsp> <Message ></pre>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Destination Access Point Identifier.</p>
Example(s)	<pre>* FETC:DATA:TEL:OTN:ODU1:TTI:DAPI:B? Returns "EXFO OTU DAPI"</pre>

:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI :OPSPec:B?

Description	<p>This query returns the generated Operator Specific.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI :OPSPec:B? <wsp> <Message></p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the generated Destination Access Point Identifier.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ODU1:TTI:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</p>

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:SAPI:EXPEcted

Description	<p>This command allows editing the expected Source Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1e/2e.</p> <p>At *RST, this value is set to EXFO ODU SAPI.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:SAPI:EXPEcted <wsp> <Message>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:E:TTI:TIM SAPI,ON</p> <p>* SENS:DATA:TEL:OTN:ODU1:E:TTI:SAPI:EXP "EXFO OTU SAPI"</p> <p>* SENS:DATA:TEL:OTN:ODU1:E:TTI:SAPI:EXP?</p> <p>Returns "EXFO OTU SAPI"</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:TTI:TIM</p> <p>* SENSe[1..n]:DATA:TELEcom:OTN:ODU1[1..n]:E:TTI:SAPI:EXPEcted?</p>

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

Description	This query returns the expected Source Access Point Identifier for ODU1e/2e. At *RST, this value is set to EXFO ODU SAPI.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:SAPI:EXPeCted? <wsp> <Message>
Parameter(s)	None
Response Syntax	<Message>

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

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the expected Source Access Point Identifier for ODU1e/2e.
Example(s)	* SENS:DATA:TEL:OTN:ODU1:E:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:ODU1:E:TTI:SAPI:EXP "EXFO OTU SAPI" * SENS:DATA:TEL:OTN:ODU1:E:TTI:SAPI:EXP? Returns "EXFO OTU SAPI"
See Also	* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:E: TTI:TIM * SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]: E:TTI:SAPI:EXPeCted

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

Description	<p>This command allows editing the expected Destination Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1e/2e.</p> <p>At *RST, this value is set to EXFO ODU DAPI.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:DAPI:EXPeCted <wsp> <Message></p>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the selected message for the instrument.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:E:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:ODU1:E:TTI:DAPI:EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:ODU1:E:TTI:DAPI:EXP? Returns "EXFO OTU DAPI"</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:TIM * SENSe[1..n]:DATA:TELEcom:OTN:ODU1[1..n]: E:TTI:DAPI:EXPeCted?</p>

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

Description	This query returns the expected Destination Access Point Identifier for ODU1e/2e. At *RST, this value is set to EXFO ODU DAPI.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:DAPI:EXPeCted? <wsp> <Message>
Parameter(s)	None
Response Syntax	<Message>

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

Response(s)

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the expected Destination Access Point Identifier for ODU1e/2e.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:E:TTI:TIM DAPI,ON

* SENS:DATA:TEL:OTN:ODU1:E:TTI:DAPI:EXP
"EXFO OTU DAPI"

* SENS:DATA:TEL:OTN:ODU1:E:TTI:DAPI:EXP?
Returns "EXFO OTU DAPI"

See Also

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

* SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]:
E:TTI:DAPI:EXPeCted

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:
TTI:SAPI:B?**

Description	<p>This query returns the generated Source Access Point Identifier for ODU1e/2e.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:SAPI:B? <wsp> <Message>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Source Access Point Identifier for ODU1e/2e.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ODU1:E:TTI:SAPI:B? Returns "EXFO OTU SAPI"</p>

:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:DAPI:B?

Description	<p>This query returns the generated Destination Access Point Identifier for ODU1e/2e.</p> <p>At *RST, this value is set to EXFO TCMi DAPI.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:DAPI:B? <wsp> <Message></p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the generated Destination Access Point Identifier for ODU1e/2e.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ODU1:E:TTI:DAPI:B? Returns "EXFO OTU DAPI"</p>

:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:OPSPec:B?

Description	<p>This query returns the generated Operator Specific for ODU1e/2e.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TTI:OPSPec:B? <wsp> <Message>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Operator Specific for ODU1e/2e.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ODU1:E:TTI:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</p>

:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI: SAPI:B16?

Description	<p>This query returns the received message of Source Access Point Identifier (SAPI) for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:SAPI:B16?</pre>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the message for the instrument.</p>
Example(s)	<pre>* FETC:DATA:TEL:OTN:OTU1:TTI:SAPI:B16? Returns "EXFO OTU SAPI"</pre>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:
DAPI:B16?**

Description	<p>This query returns the received message of Destination Access Point Identifier (DAPI) for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:DAPI:B16?
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the message for the instrument.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OTU1:TTI:DAPI:B16?</p> <p>Returns "EXFO OTU DAPI"</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:
OPSPec:B32?**

Description	<p>This query returns the received message of Operator Specific for the instrument.</p> <p>At *RST, this value of OPSpecific set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:OPSPec:B32?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the message for the instrument.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OTU1:TTI:OPSP:B32? Returns "EXFO OTU OPERATOR SPECIFIC"</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TCM[1..n]:TTI:SAPI:B?**

Description	<p>This query returns the generated Source Access Point Identifier for the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:SAPI:B?</code>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Source Access Point Identifier.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ODU1:TCM1:TTI:SAPI:B? Returns "EXFO OTU SAPI"</p>

:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:DAPI:B?

Description	<p>This query returns the generated Destination Access Point Identifier of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi DAPI.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:DAPI:B?</pre>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Destination Access Point Identifier.</p>
Example(s)	<pre>* FETC:DATA:TEL:OTN:ODU1:TCM1:TTI:DAPI:B? Returns "EXFO OTU DAPI"</pre>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TCM[1..n]:TTI:OPSPec:B?**

Description	<p>This query returns the generated Operator Specific of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:OPSPec:B?
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the generated Operator Specific.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ODU1:TCM1:TTI:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</p>

:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:SAPI:B?

Description	<p>This query returns the received message for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:TCM[1..n]:TTI:SAPI:B? <wsp></code>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OTU1:TTI:SAPI:B16? Returns "EXFO OTU SAPI"</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:
TCM[1..n]:TTI:DAPI:B?**

Description	<p>This query returns the received message for the instrument.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:DAPI:B? <wsp></pre>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<pre>* FETC:DATA:TEL:OTN:OTU1:TTI:DAPI:B1? Returns "EXFO OTU DAPI"</pre>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:
TCM[1..n]:TTI:OPSPec:B?**

Description	<p>This query returns the received message for the instrument.</p> <p>At *RST, this value of OPSpecific set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: TCM[1..n]:TTI:OPSPec:B? <wsp></p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OTU1:TTI:OPSP:B32? Returns "EXFO OTU OPERATOR SPECIFIC"</p>

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

This command enables or disables the Trace Identifier Mismatch (TIM).

At *RST, this value is set to ON.

Syntax

:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
TTI:TIM<wsp>SAPI|DAPI,<Set>

Parameter(s)

Etim:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SAPI|DAPI.

Enables or disables the TIM (Trace Identifier Mismatch).

SAPI, selects the SAPI which allows editing of the Source Access Point Identifier (SAPI) message to be generated.

DAPI, selects the DAPI which allows editing of the Destination Access Point Identifier (DAPI) message to be generated.

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

Set:

The program data syntax for the second parameter is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Trace Identifier Mismatch (TIM).

Example(s)

* SENS:DATA:TEL:OTN:OTU1:TTI:TIM SAPI,ON
* SENS:DATA:TEL:OTN:OTU1:TTI:TIM? SAPI
Returns 1

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

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

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

Description	<p>This query returns status of Trace Identifier Mismatch (TIM).</p> <p>At *RST, this value is set to ON.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:TIM? <wsp>SAPI DAPI</p>
Parameter(s)	<p>Etim:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SAPI DAPI.</p> <p>Enables or disables the TIM (Trace Identifier Mismatch).</p> <p>SAPI, selects the SAPI which allows editing of the Source Access Point Identifier (SAPI) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the Destination Access Point Identifier (DAPI) message to be generated.</p>
Response Syntax	<p><Set></p>

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

Response(s)	Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Trace Identifier Mismatch (TIM).
Example(s)	* SENS:DATA:TEL:OTN:OTU1:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:OTU1:TTI:TIM? SAPI Returns 1
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:TTI:TIM

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

This command enables or disables the state of TCM-Trace Identifier Mismatch (TIM) for the instrument.

At *RST, this value is set to ON.

Syntax

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TCM[1..n]:TTI:TIM<wsp>SAPI|DAPI, <Set>

:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: TCM[1..n]:TTI:TIM

Parameter(s)	<p>Etim:</p> <p>The program data syntax for <Etim> is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Trail Trace Identifier (TTI) Overwrite feature.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:TIM SAPI,ON</p> <p>* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:TIM? SAPI Return 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: TCM[1..n]TTI:TIM?</p>

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

Description	<p>This query returns state of TCM-Trace Identifier Mismatch (TIM) for the instrument.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSE[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TCM[1..n]:TTI:TIM? <wsp>SAPI DAPI</pre>
Parameter(s)	<p>Etim:</p> <p>The program data syntax for <Etim> is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
Response Syntax	<pre><Set></pre>

:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: TCM[1..n]:TTI:TIM?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of Trace Identifier Mismatch (TIM).

SAPI, SAPI (Source Access Point Identifier) is selected as the message to be generated.

DAPI, DAPI (Source Access Point Identifier) is selected as the message to be generated.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:TIM
SAPI,ON

* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:TIM?
SAPI Return 1

See Also

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

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

This command enables or disables the state of Trace Identifier Mismatch (TIM) for the instrument.

At *RST, this value is set to ON.

Syntax

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TTI:TIM<wsp>SAPI|DAPI, <Set>

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

Parameter(s)	<p>Etim:</p> <p>The program data syntax for <Etim> is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Trail Trace Identifier (TTI) Overwrite feature.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:TTI:TIM SAPI,ON</p> <p>* SENS:DATA:TEL:OTN:ODU1:TTI:TIM? SAPI</p> <p>Return 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:TTI:TIM?</p>

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

Description	<p>This query returns state of Trace Identifier Mismatch (TIM) for the instrument.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TTI:TIM? <wsp>SAPI DAPI</pre>
Parameter(s)	<p>Etim:</p> <p>The program data syntax for <Etim> is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
Response Syntax	<Set>

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

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of Trace Identifier Mismatch (TIM).

SAPI, SAPI (Source Access Point Identifier) is selected as the message to be generated.

DAPI, DAPI (Source Access Point Identifier) is selected as the message to be generated.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:TTI:TIM SAPI,ON

* SENS:DATA:TEL:OTN:ODU1:TTI:TIM? SAPI

Return 1

See Also

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

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

This command enables or disables the state of TCM-Trace Identifier Mismatch (TIM) for the instrument for ODU1e/2e.

At *RST, this value is set to ON.

Syntax

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:TCM[1..n]:TTI:TIM<wsp>SAPI|DAPI, <Set>

:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: E:TCM[1..n]:TTI:TIM

Parameter(s)	<p>Etim:</p> <p>The program data syntax for <Etim> is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Trail Trace Identifier (TTI) Overwrite feature.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:TIM SAPI,ON</p> <p>* SENS:DATA:TEL:OTN:ODU1:TCM1:TTI:TIM? SAPI Return 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: TCM[1..n]TTI:TIM?</p>

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

Description	<p>This query returns state of TCM-Trace Identifier Mismatch (TIM) for the instrument for ODU1e/2e.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TCM[1..n]:TTI:TIM?<wsp>SAPI DAPI</pre>
Parameter(s)	<p>Etim:</p> <p>The program data syntax for <Etim> is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
Response Syntax	<Set>

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

Response(s)

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of Trace Identifier Mismatch (TIM).

SAPI, SAPI (Source Access Point Identifier) is selected as the message to be generated.

DAPI, DAPI (Source Access Point Identifier) is selected as the message to be generated.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:TIM
SAPI,ON

* SENS:DATA:TEL:OTN:ODU1:E:TCM1:TTI:TIM?
SAPI Return 1

See Also

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

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

This command enables or disables the state of Trace Identifier Mismatch (TIM) for the instrument for ODU1e/2e.

At *RST, this value is set to ON.

Syntax

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:TTI:TIM<wsp>SAPI|DAPI, <Set>

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

Parameter(s)	<p>Etim:</p> <p>The program data syntax for <Etim> is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Trail Trace Identifier (TTI) Overwrite feature.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:E:TTI:TIM SAPI,ON</p> <p>* SENS:DATA:TEL:OTN:ODU1:E:TTI:TIM? SAPI</p> <p>Return 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:TTI:TIM?</p>

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

Description	<p>This query returns state of Trace Identifier Mismatch (TIM) for the instrument for ODU1e/2e.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:TIM?<wsp>SAPI DAPI</pre>
Parameter(s)	<p>Etim:</p> <p>The program data syntax for <Etim> is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
Response Syntax	<Set>

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

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of Trace Identifier Mismatch (TIM).

SAPI, SAPI (Source Access Point Identifier) is selected as the message to be generated.

DAPI, DAPI (Source Access Point Identifier) is selected as the message to be generated.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:E:TTI:TIM SAPI,ON

* SENS:DATA:TEL:OTN:ODU1:E:TTI:TIM? SAPI
Return 1

See Also

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

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
CONFig:TCM[1..n]**

Description	<p>This command enables or disables the configuration of TCM.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: CONFig:TCM[1..n]<wsp><Level></pre>
Parameter(s)	<p>Level:</p> <p>The program data syntax for <Level> is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Enables or disables the configuration of TCM.</p>
Example(s)	<pre>* SOURce:DATA:TEL:OTN:ODU1:E:CONF:TCM1 ON * SOURce:DATA:TEL:OTN:ODU1:E:CONF:TCM1? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:CONF:TCM[1..n]?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: CONFig:TCM[1..n]?

Description	This query returns the status of the configuration of TCM. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: CONFig:TCM[1..n]? <wsp> <Level>
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
CONFig:TCM[1..n]?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the configuration of TCM for ODU1e/2e.
Example(s)	* SOURce:DATA:TEL:OTN:ODU1:E:CONF:TCM1 ON * SOURce:DATA:TEL:OTN:ODU1:E:CONF:TCM1? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:CONF:TCM[1..n]?

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: :CONFig:TCM[1..n]

Description	<p>This command enables or disables the configuration of TCM for ODU1e/2e.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:CONFig:TCM[1..n]<wsp><Level></pre>
Parameter(s)	<p>Level:</p> <p>The program data syntax for <Level> is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Enables or disables the configuration of TCM for ODU1e/2e.</p>
Example(s)	<pre>* SOURce:DATA:TEL:OTN:ODU1:E:CONF:TCM1 ON * SOURce:DATA:TEL:OTN:ODU1:E:CONF:TCM1? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:CONF:TCM[1..n]?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E
:CONFig:TCM[1..n]?**

Description	This query returns the status of the configuration of TCM for ODU1e/2e. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:CONFig:TCM[1..n]?<wsp><Level>
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E :CONF:TCM[1..n]?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of the configuration of TCM for ODU1e/2e.

Example(s)

* SOURce:DATA:TEL:OTN:ODU1:E:CONF:TCM1
ON

* SOURce:DATA:TEL:OTN:ODU1:E:CONF:TCM1?
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:CONF:TCM[1..n]?

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TTI:OVERwrite:ENABled**

Description	<p>This command enables or disables the Trail Trace Identifier (TTI) Overwrite feature for standard rates OTU1/2.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TTI:OVERwrite:ENABled<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Trail Trace Identifier (TTI) Overwrite feature.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:TTI:OVER:ENAB ON * SOUR:DATA:TEL:OTN:ODU1:TTI:OVER:ENAB? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TTI:OVERwrite:ENABled?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TTI:OVERwrite:ENABLEd?**

Description This query returns the status of Trail Trace Identifier (TTI) Overwrite feature for standard rates OTU1/2.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TTI:OVERwrite:ENABLEd?

Parameter(s) None

Response Syntax <Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
TTI:OVERwrite:ENABLEd?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Trail Trace Identifier (TTI) Overwrite feature.
Example(s)	* SOUR:DATA:TEL:OTN:ODU1:TTI:OVER:ENAB ON * SOUR:DATA:TEL:OTN:ODU1:TTI:OVER:ENAB? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: TTI:OVERwrite:ENABLEd

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

Description This command sets the Fault Type Fault Location (FTFL) Fault Indicator message to be generated for transmitter.

At *RST, this value is set to NFAult.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:INDication<wsp>FORWard|BACKward,
NFAult|SFAil|SDEGrade|REServed

Parameter(s) Ftfl:
The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWard|BACKward.
Sets the configuration of the forward and backward ODU (Optical Data Unit) FTFL (Fault Type Fault Location) to be generated.
FORWard, sets the Forward configuration.
BACKward, sets the Backward configuration.

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

Indication:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

NFAult | SFAil | SDEGrade | REServed.

Sets the Fault Type Fault Location (FTFL) Fault Indicator message to be generated.

NFAult, selects the NFAult (No Fault) as Fault Indicator.

SFAil, selects the SFAil (Signal Fail) as Fault Indicator.

SDEGrade, selects the SDEGrade (Signal Degraded) as Fault Indicator.

REServed, selects the Reserved as Fault Indicator.

Example(s)

* SOUR:DATA:TEL:OTN:ODU1:FTFL:IND
FORW,SFA

* SOUR:DATA:TEL:OTN:ODU1:FTFL:IND? FORW
Returns SFAIL

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

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

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:INDIcation?**

Description This query returns the Fault Type Fault Location (FTFL) Fault Indicator message to be generated for transmitter.

At *RST, this value is set to NFAult.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:INDIcation?<wsp>FORWard|BACKward

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWard|BACKward.
Sets the configuration of the forward and backward ODU (Optical Data Unit) FTFL (Fault Type Fault Location) to be generated.
FORWard, sets the Forward configuration.
BACKward, sets the Backward configuration.

Response Syntax <Indication>

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

Response(s)	<p>Indication:</p> <p>The response data syntax for <Indication> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the Fault Type Fault Location (FTFL) Fault Indicator message to be generated.</p> <p>NFAULT, No Fault (NFAULT) is selected as Fault Type Fault Location (FTFL) Fault Indicator.</p> <p>SFAIL, Signal Fail (SFAIL) is selected as Fault Type Fault Location (FTFL) Fault Indicator.</p> <p>SDEGRADE, Signal Degraded (SDEGRADE) is selected as Fault Type Fault Location (FTFL) Fault Indicator.</p> <p>RESERVED, Reserved is selected as Fault Type Fault Location (FTFL) Fault Indicator.</p>
Example(s)	<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>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:INDication</p>

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

Description	<p>This command sets the FTFL (Fault Type Fault Location) Fault Indicator Code to be generated for transmitter.</p> <p>At *RST, this value is set to #H00.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:CODE<wsp>FORWard BACKward, <Code></pre>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FORWard BACKward.</p> <p>Sets the configuration of the forward and backward ODU FTFL to be generated.</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>

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

The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the of the FTFL Fault Indicator Code to be generated.

The values are #H00 through #HFF.

Example(s)

* SOUR:DATA:TEL:OTN:ODU1:FTFL:CODE
FORW,#H01
* SOUR:DATA:TEL:OTN:ODU1:FTFL:CODE?
FORW Returns #H01

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

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

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

Description	<p>This query returns the FTFL Fault Indicator Code to be generated for transmitter.</p> <p>At *RST, this value is set to #H00.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:CODE? <wsp>FORWard BACKward</p>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FORWard BACKward.</p> <p>Sets the configuration of the forward and backward ODU FTFL to be generated.</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>
Response Syntax	<p><Code></p>

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

Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the Fault Indicator Code to be generated. The values are #H00 through #HFF.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:FTFL:CODE FORW,#H01</p> <p>* SOUR:DATA:TEL:OTN:ODU1:FTFL:CODE? FORW Returns #H01</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:CODE</p>

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

Description This command sets the FTFL (Fault Type Fault Location) Operator Identifier to be generated for transmitter.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:IDENtifier<wsp>FORWard|BACKward,
<Identifier>

Parameter(s) Fttl:
The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWard|BACKward.
Sets the configuration of the forward and backward ODU FTFL to be generated.
FORWard, sets the Forward configuration.
BACKward, sets the Backward configuration.

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

The program data syntax for the second parameter is defined as a <STRING PROGRAM DATA> element.

Sets the Operator Identifier to be generated.

Choices are 1 to 9 bytes for forward and 129 to 137 bytes for backward. A maximum of 9 characters are allowed.

Example(s)

* SOUR:DATA:TEL:OTN:ODU1:FTFL:IDEN
FORW,"exfo"

* SOUR:DATA:TEL:OTN:ODU1:FTFL:IDEN?
FORW Returns "exfo"

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

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

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:IDENTifier?**

Description This query returns the FTFL Operator Identifier to be generated for transmitter.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:IDENTifier?<wsp>FORWard|BACKward

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWard|BACKward.
Sets the configuration of the forward and backward ODU FTFL to be generated.
FORWard, sets the Forward configuration.
BACKward, sets the Backward configuration.

Response Syntax <Identifier>

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

Response(s)	<p>Identifier:</p> <p>The response data syntax for <Identifier> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the value of the Operator Identifier to be generated. Choices are 1 to 9 bytes for forward and 129 to 137 bytes for backward. A maximum of 9 characters are allowed.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:FTFL:IDEN FORW,"exfo"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:FTFL:IDEN? FORW Returns "exfo"</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:IDENtifier</p>

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

Description	<p>This command sets the FTFL (Fault Type Fault Location) Operator Specific to be generated for transmitter.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OPSPec<wsp>FORWard BACKward, <Specific></pre>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FORWard BACKward.</p> <p>Sets the configuration of the forward and backward ODU FTFL to be generated.</p> <p>FORWard, sets the Forward configuration.</p> <p>BACKward, sets the Backward configuration.</p>

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

The program data syntax for the second parameter is defined as a <STRING PROGRAM DATA> element.

Sets the Operator Specific to be generated.

Choices are 10 to 127 bytes for forward and 138 to 225 bytes for backward. A maximum of 118 characters are allowed.

Example(s)

* SOUR:DATA:TEL:OTN:ODU1:FTFL:OPSP
FORW,"exfo"
* SOUR:DATA:TEL:OTN:ODU1:FTFL:OPSP?
FORW Returns "exfo"

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

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

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

Description This query returns the FTFL Operator Specific to be generated for transmitter.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:OPSPec? <wsp>FORWARD|BACKWARD

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWARD|BACKWARD.
Sets the configuration of the forward and backward ODU FTFL to be generated.
FORWARD, sets the Forward configuration.
BACKWARD, sets the Backward configuration.

Response Syntax <Specific>

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

Response(s)	<p>Specific: The response data syntax for <Specific> is defined as a <STRING RESPONSE DATA> element. Returns the value of Operator Specific. Choices are 10 to 127 bytes for forward and 138 to 225 bytes for backward. A maximum of 118 characters are allowed.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:FTFL:OPSP FORW,"exfo" * SOUR:DATA:TEL:OTN:ODU1:FTFL:OPSP? FORW Returns "exfo"</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:FTFL:OPSPec</p>

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:INDIcation?

Description	<p>This query returns the FTFL (Fault Type Fault Location) Fault Indicator to be generated for receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:INDIcation?<wsp>FORWard BACKward</pre>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FORWard BACKward.</p> <p>Sets the configuration of the forward and backward ODU FTFL Fault Indicator to be generated.</p> <p>FORWard, sets the Forward configuration. BACKward, sets the Backward configuration.</p>
Response Syntax	<pre><Indication></pre>

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

Response(s)	<p>Indication:</p> <p>The response data syntax for <Indication> is defined as a <CHARACTER RESPONSE DATA > element.</p> <p>Returns the FTFL Fault Indicator to be generated. NFAULT, No Fault (NFAULT) is selected as FTFL Fault Indicator.</p> <p>SFAIL, Signal Fail (SFAIL) is selected as FTFL Fault Indicator.</p> <p>SDEGRADE, Signal Degraded (SDEGRADE) is selected as FTFL Fault Indicator.</p> <p>RESERVED, Reserved is selected as FTFL Fault Indicator.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:FTFL:IND? FORW</p> <p>Returns the FTFL Fault Indicator message to be generated.</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>

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

Description This query returns the FTFL Fault Indicator Code to be generated for receiver.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:CODE? <wsp>FORWard|BACKward

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWard|BACKward.
Sets the configuration of the forward and backward ODU FTFL to be generated.
FORWard, sets the Forward configuration.
BACKward, sets the Backward configuration.

Response Syntax <Code>

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

Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the FTFL Fault Indicator Code to be generated.</p> <p>The values are #H00 through #HFF.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:FTFL:CODE? FORW Returns the Fault Indication code to be generated.</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>

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

Description This query returns the FTFL (Fault Type Fault Location) Operator Identifier to be generated for receiver.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:IDENtifier?<wsp>FORWard|BACKward

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWard|BACKward.
Sets the configuration of the forward and backward ODU FTFL to be generated.
FORWard, sets the Forward configuration.
BACKward, sets the Backward configuration.

Response Syntax <Identifier>

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:IDENTifier?**

Response(s)	Identifier: The response data syntax for <Identifier> is defined as a <STRING RESPONSE DATA> element. Returns the value of the Operator Identifier (bytes 1 to 9 for forward, byte 129 to 137 for backward) to be generated.
Example(s)	* SENS:DATA:TEL:OTN:ODU1:FTFL:IDEN? FORW Returns the FTFL Operator Identifier to be generated.
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:SPECific?

Description This query returns the FTFL (Fault Type Fault Location) Operator Specific to be generated for receiver.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:SPECific? <wsp>FORWard|BACKward

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWard|BACKward.
Sets the configuration of the forward and backward ODU FTFL to be generated.
FORWard, sets the Forward configuration.
BACKward, sets the Backward configuration.

Response Syntax <Specific>

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:SPECific?**

Response(s)	Specific: The response data syntax for <Specific> is defined as a <STRING RESPONSE DATA> element. Returns the value of Operator Specific (bytes 10 to 127 for forward, byte 138 to 255 for backward) to be generated.
Example(s)	* SENS:DATA:TEL:OTN:ODU1:FTFL:SPEC? FORW Returns the FTFL Operator Specific to be generated.
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OVERwrite:ENABLEd

Description	<p>This command enables or disables the Fault Type Fault Location (FTFL) Overwrite feature for standard rates OTU1/2.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OVERwrite:ENABLEd <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Fault Type Fault Location (FTFL) Overwrite feature.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:FTFL:OVER:ENAB ON * SOUR:DATA:TEL:OTN:ODU1:FTFL:OVER: ENAB? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OVERwrite:ENABLEd?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
FTFL:OVERwrite:ENABLEd?**

Description	This query returns the status of Fault Type Fault Location (FTFL) Overwrite feature for standard rates OTU1/2. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OVERwrite:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OVERwrite:ENABLEd?

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Fault Type Fault Location (FTFL) Overwrite feature.
Example(s)	* SOUR:DATA:TEL:OTN:ODU1:FTFL:OVER:ENAB ON * SOUR:DATA:TEL:OTN:ODU1:FTFL:OVER: ENAB? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: FTFL:OVERwrite:ENABLEd

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

Description

This command sets the injected payload signal type to be generated for transmitter.

At *RST, this value is set to PRBStest.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
PTYPe<wsp>EXPerimental|ASYNchronous|
BISynch|ATM|GFP|VCONcate|BSTiming|
BSNTiming|ODUMux|RFSTandard|RPRopriet|
NULLtest|PRBStest|NAVailable|1000BODU0|FC
1200ODU2E|GFPEOPU2|OC3STM1ODU0|OC12S
TM4ODU0|FC100ODU0|FC200ODU1|FC400ODU
FLEX|FC800ODUFLEX|ODUODTUKTS|
ODUODTUJK
```

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPE

Parameter(s)	<p>Payload:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>EXPerimental ASYNchronous BISYNch ATM GFP VCONcate BSTiming BSNTiming RFSTandard RPRopriet NULLtest PRBStest NAVailable 1000BODU0 FC1200ODU2E GFPEOPU2 OC3STM1ODU0 OC12STM4ODU0 FC100ODU0 FC200ODU1 FC400ODUFLEX FC800ODUFLEX ODUODTUKTS ODUODTUJK.</p> <p>Sets the expected payload signal type to be generated.</p> <p>EXPerimental, selects the payload type as Experimental.</p> <p>ASYNchronous, selects the payload type as Asynchronous.</p> <p>BISYNch, selects the payload type as Bit Synchronous.</p> <p>ATM, selects the payload type as Asynchronous Transfer Mode (ATM).</p> <p>GFP, selects the payload type as Generic Framing Procedure (GFP).</p> <p>VCONcate, selects the payload type as Virtual Concatenation.</p> <p>BSTiming, selects the payload type as Bit Stream Timing.</p>
---------------------	--

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
PTYPE**

BSNTiming, selects the payload type as Bit Stream No Timing.

RFStandard, selects the payload type as Reserved Future Standardization.

RPRopriet, selects the payload type as Reserved Proprietary.

NULLtest, selects the payload type as NULL Test.

PRBStest, selects the payload type as Pseudo Random Bit Sequence (PRBS) Test.

NAVailable, selects the payload type as Not Available.

1000BODU0, selects the payload type as 1000BODU0.

FC1200ODU2, selects the payload type as FC1200ODU2.

EGFPEOPU2, selects the payload type as EGFPEOPU2.

OC3STM1ODU0, selects the payload type as OC3STM1ODU0.

OC12STM4ODU0, selects the payload type as OC12STM4ODU0.

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

FC100ODU0, selects the payload type as FC100ODU0.

FC200ODU1, selects the payload type as FC200ODU1.

FC400ODUFLEX, selects the payload type as FC400ODUFLEX.

FC800ODUFLEX, selects the payload type as FC800ODUFLEX.

ODUODTUKTS, selects the payload type as ODUODTUKTS.

ODUODTUJK, selects the payload type as ODUODTUJK.

Example(s)

* SOUR:DATA:TEL:OTN:OPU1:PTYP EXP
* SOUR:DATA:TEL:OTN:OPU1:PTYP?
Returns EXPERIMENTAL

Note

* OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
* The injected payload type is automatically updated when the code field is changed and vice versa.

See Also

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

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPE?

Description	<p>This query returns the injected payload signal type to be generated for transmitter.</p> <p>At *RST, this value is set to PRBStest.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPE?
Parameter(s)	None
Response Syntax	<Payload>
Response(s)	<p>Payload:</p> <p>The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the injected payload signal type to be generated</p> <p>EXPERIMENTAL, Experimental is selected as payload type.</p> <p>ASYNCHRONOUS, Asynchronous is selected as payload type.</p> <p>BISYNCH, Bit Synchronous is selected as payload type.</p>

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPE?

ATM, Asynchronous Transfer Mode (ATM) is selected payload type.

GFP, Generic Framing Procedure (GFP) is selected as payload type.

VCONCATENATE, Virtual Concatenation is selected as payload type.

BSTIMING, Bit Stream Timing is selected as payload type.

BSNTIMING, Bit Stream No Timing is selected as payload type.

RFSTANDARD, Reserved Future Standardization is selected as payload type.

RPROPRIET, Reserved Proprietary is selected as payload type.

NULLTEST, NULL Test is selected as payload type.

PRBSTEST, Pseudo Random Bit Sequence (PRBS) Test is selected as payload type.

NAVAILABLE, Not available is selected as payload type.

1000BODU0,1000BODU0 is selected as payload type.

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

FC1200ODU2, FC1200ODU2 is selected as payload type.

EGFPEOPU2, EGFPEOPU2 is selected as payload type.

OC3STM1ODU0, OC3STM1ODU0 is selected as payload type.

OC12STM4ODU0, OC12STM4ODU0 is selected as payload type.

FC100ODU0, FC100ODU0 is selected as payload type.

FC200ODU1, FC200ODU1 is selected as payload type.

FC400ODUFLEX, FC400ODUFLEX is selected as payload type.

FC800ODUFLEX, FC800ODUFLEX is selected as payload type.

ODUODTUKTS, ODUODTUKTS is selected as payload type.

ODUODTUJK, ODUODTUJK is selected as payload type.

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

Example(s)

* SOUR:DATA:TEL:OTN:OPU1:PTYP EXP
* SOUR:DATA:TEL:OTN:OPU1:PTYP?
Returns EXPERIMENTAL

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

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

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

Description	This query returns the received payload signal type to be generated for the receiver. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPe:RECeived?
Parameter(s)	None
Response Syntax	<Payload>

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

Response(s)

Payload:

The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the received payload type.

Returns the received payload type.

EXPERIMENTAL, Experimental is retrieved.

ASYNCHRONOUS, Asynchronous is retrieved.

BISYNCH, Bit Synchronous (BISYNCH) is retrieved.

ATM, Asynchronous Transfer Mode (ATM) is retrieved.

GFP1, Generic Framing Procedure (GFP) is retrieved.

VCONCATENATE, Virtual Concatenation (VCONCATENATE) is retrieved.

BSTIMING, Bit Stream Timing (BSTIMING) is retrieved.

BSNTIMING, Bit Stream No Timing (BSNTIMING) is retrieved.

ODUMUX, Optical Data Unit (ODU) Mux is retrieved.

RFSTANDARD, Reserved Future Standardization (RFSTANDARD) is retrieved.

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

RPROPRIET, Reserved Proprietary (RPROPRIET) is retrieved.

NULLTEST, NULL Test is retrieved.

PRBSTEST, Pseudo Random Bit Sequence Test (PRBSTEST) is retrieved.

NAVAILABLE, Not available is retrieved.

Example(s)

* FETC:DATA:TEL:OTN:OPU1:PTYP:REC? Returns the received payload type.

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

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

Description This command sets the expected payload signal type to be generated for receiver.

At *RST, this value is set to PRBStest.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
PTYPe <wsp> EXPerimental | ASYNchronous |
BISYNch | ATM | GFP1 | VCONcate | BSTiming |
BSNTiming | ODUMux | RFSTandard | RPRopriet |
NULLtest | PRBStest | NAVailable | | 1000BODU0 | F
C1200ODU2E | GFPEOPU2 | OC3STM1ODU0 | OC1
2STM4ODU0 | FC100ODU0 | FC200ODU1 | FC400O
DUFLEX | FC800ODUFLEX | ODUODTUKTS | ODU
ODTUJK.

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

Parameter(s)

Payload:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

EXPerimental|ASYNchronous|BISYNch|ATM|
GFP1|VCONcate|BSTiming|BSNTiming|
ODUMux|RFSTandard|RPRopriet|NULLtest|
PRBStest|NAVailable||1000BODU0|FC1200ODU
2E|GFPEOPU2|OC3STM1ODU0|OC12STM4ODU
0|FC100ODU0|FC200ODU1|FC400ODUFLEX|FC
800ODUFLEX|ODUODTUKTS|ODUODTUJK.

Sets the expected payload signal type to be generated.

EXPerimental, selects the payload type as Experimental.

ASYNchronous, selects the payload type as Asynchronous.

:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]: PTYPe

BISynch, selects the payload type as Bit Synchronous.

ATM, selects the payload type as Asynchronous Transfer Mode (ATM).

GFP1, selects the payload type as Generic Framing Procedure (GFP).

VCONcate, selects the payload type as Virtual Concatenation.

BSTiming, selects the payload type as Bit Stream Timing.

BSNTiming, selects the payload type as Bit Stream No Timing.

ODUMux, selects the payload type as Optical Data Unit (ODU) Mux.

RFStandard, selects the payload type as Reserved Future Standardization.

RPRopriet, selects the payload type as Reserved Proprietary.

NULLtest, selects the payload type as NULL Test.

PRBStest, selects the payload type as Pseudo Random Bit Sequence (PRBS) Test.

NAvailable, selects the payload type as Not Available.

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

000BODU0, selects the payload type as
000BODU0.

FC1200ODU2, selects the payload type as
FC1200ODU2.

EGFPEOPU2, selects the payload type as
EGFPEOPU2.

OC3STM1ODU0, selects the payload type as
OC3STM1ODU0.

OC12STM4ODU0, selects the payload type as
OC12STM4ODU0.

FC100ODU0, selects the payload type as
FC100ODU0.

FC200ODU1, selects the payload type as
FC200ODU1.

FC400ODUFLEX, selects the payload type as
FC400ODUFLEX.

FC800ODUFLEX, selects the payload type as
FC800ODUFLEX.

ODUODTUKTS, selects the payload type as
ODUODTUKTS.

ODUODTUJK, selects the payload type as
ODUODTUJK.

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

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:OTN:OPU1:PTYP EXP* SENS:DATA:TEL:OTN:OPU1:PTYP? Returns EXPERIMENTAL
Note	<ul style="list-style-type: none">* OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.* The injected payload type is automatically updated when the code field is changed and vice versa.
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PTYPe?

**:SENSe[1..n]:DATA:TELeom:OTN:OPU[1..n]:
PTYPE?**

Description	<p>This query returns the expected payload signal type to be generated for receiver.</p> <p>At *RST, this value is set to PRBStest.</p>
Syntax	:SENSe[1..n]:DATA:TELeom:OTN:OPU[1..n]: PTYPE?
Parameter(s)	None
Response Syntax	<Payload>
Response(s)	<p>Payload:</p> <p>The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the injected payload signal type to be generated</p> <p>EXPERIMENTAL, Experimental is selected as payload type.</p> <p>ASYNCHRONOUS, Asynchronous is selected as payload type.</p> <p>BISYNCH, Bit Synchronous is selected as payload type.</p> <p>ATM, Asynchronous Transfer Mode (ATM) is selected as payload type.</p>

:SENSe[1..n]:DATA:TELeom:OTN:OPU[1..n]:PTYPE?

GFP1, Generic Framing Procedure (GFP) is selected as payload type.

VCONCATENATE, Virtual Concatenation is selected as payload type.

BSTIMING, Bit Stream Timing is selected as payload type.

BSNTIMING, Bit Stream No Timing is selected as payload type.

ODUMUX, Optical Data Unit (ODU) Mux is selected as payload type.

RFSTANDARD, Reserved Future Standardization is selected as payload type.

RPROPRIET, Reserved Proprietary is selected as payload type.

NULLTEST, NULL Test is selected as payload type.

PRBSTEST, Pseudo Random Bit Sequence (PRBS) Test is selected as payload type.

NAVAILABLE, Not available is selected as payload type.

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

1000BODU0, 000BODU0 is selected as payload type.

FC1200ODU2, FC1200ODU2 is selected as payload type.

EGFPEOPU2, EGFPEOPU2 is selected as payload type.

OC3STM1ODU0, OC3STM1ODU0 is selected as payload type.

OC12STM4ODU0, OC12STM4ODU0 is selected as payload type.

FC100ODU0, FC100ODU0 is selected as payload type.

FC200ODU1, FC200ODU1 is selected as payload type.

FC400ODUFLEX, FC400ODUFLEX is selected as payload type.

FC800ODUFLEX, FC800ODUFLEX is selected as payload type.

ODUODTUKTS, ODUODTUKTS is selected as payload type.

ODUODTUJK, ODUODTUJK is selected as payload type.

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

Example(s) * SENS:DATA:TEL:OTN:OPU1:PTYP EXP
 * SENS:DATA:TEL:OTN:OPU1:PTYP?
 Returns EXPERIMENTAL

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:
 PTYPe

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
PTYPe:OVERwrite:ENABLEd**

Description This command enables or disables the Payload Type Overwrite feature for standard rates OTU1/2.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
PTYPe:OVERwrite:ENABLEd <wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPe:OVERwrite:ENABLEd

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Payload Type Overwrite feature.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OPU1:PTYP:OVER:ENAB ON</p> <p>* SOUR:DATA:TEL:OTN:OPU1:PTYP:OVER: ENAB? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPe:OVERwrite:ENABLEd?</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
PTYPE:OVERwrite:ENABLEd?**

Description	<p>This query returns the status of Payload Type Overwrite feature for standard rates OTU1/2.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPE:OVERwrite:ENABLEd?</pre>
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the Payload Type Overwrite feature.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OPU1:PTYP:OVER:ENAB ON * SOUR:DATA:TEL:OTN:OPU1:PTYP:OVER: ENAB? Returns 1</pre>
See Also	<pre>* SOURCE[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PTYPE:OVERwrite:ENABLEd</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE

Description This command sets the code of the payload type for transmitter.

At *RST, this value is set to #H03.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
PCODE<wsp><Code>

Parameter(s) Code:
The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.
Sets the code of the payload type.
The values are #H00 through #HFF.

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
PCODE****Example(s)**

* SOUR:DATA:TEL:OTN:OPU1:PCOD #H00
* SOUR:DATA:TEL:OTN:OPU1:PCOD?
Returns #H00

Note

* OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
* The code field is automatically updated when the injected payload type is changed and vice versa.

See Also

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

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

Description	<p>This query returns the code of the payload type for transmitter.</p> <p>At *RST, this value is set to #H03.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Code></p>
Response(s)	<p>Code: The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the corresponding injected payload type as hexadecimal code. The values are #H00 through #HFF.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
PCODE?****Example(s)**

* SOUR:DATA:TEL:OTN:OPU1:PCOD #H00
* SOUR:DATA:TEL:OTN:OPU1:PCOD?
Returns #H00

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
PCODE

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

Description	<p>This query returns the received code of the payload type.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE:RECEived?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Code></p>
Response(s)	<p>Code: The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the received payload code.</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OPU1:PCOD:REC? #H00 Returns the received payload code.</p>

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

Description	<p>This command sets the code of the payload type for receiver.</p> <p>At *RST, this value is set to #H03.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE<wsp><Code></pre>
Parameter(s)	<p>Code:</p> <p>The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the corresponding expected payload type as hexadecimal code.</p> <p>The values are #H00 through #HFF.</p>

:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:OTN:OPU1:PCOD #H00* SENS:DATA:TEL:OTN:OPU1:PCOD? Returns #H00
Note	<ul style="list-style-type: none">* OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.* The code field is automatically updated when the injected payload type is changed and vice versa.
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PCODE?

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

Description	<p>This query returns the code of the payload type for receiver.</p> <p>At *RST, this value is set to #H03.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PCODE?
Parameter(s)	None
Response Syntax	<Code>
Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the corresponding expected payload type as hexadecimal code.</p> <p>The values are #H00 through #HFF.</p>

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

Example(s)

* SENS:DATA:TEL:OTN:OPU1:PCOD #H00
* SENS:DATA:TEL:OTN:OPU1:PCOD?
Returns #H00

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

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

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

Description	<p>This command enables or disables the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.</p> <p>At *RST this value is set to OFF.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PLM<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the OPU-PLM (Optical Payload Unit-Payload Label Mismatch) alarm analysis.</p>

**:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:
PLM**

Example(s) * SENS:DATA:TEL:OTN:OPU1:PLM ON
 * SENS:DATA:TEL:OTN:OPU1:PLM? Returns 1

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:
 PLM?

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

Description	<p>This query returns the status of the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.</p> <p>At *RST this value is set to OFF.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PLM?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.</p>

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

Example(s) * SENS:DATA:TEL:OTN:OPU1:PLM ON
 * SENS:DATA:TEL:OTN:OPU1:PLM? Returns 1

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:
 PLM

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:TYPE

Description	<p>This command selects the type of Optical Transport Unit (OTU) alarm.</p> <p>At *RST, this value is set to OAIS.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:TYPE<wsp>OAIS OBDI LOF2 OOF1 LOM OOM OBlae OIAE</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OAIS OBDI LOF2 OOF1 LOM OOM OBlae OIAE.</p> <p>Selects the type of OTU (Optical Transport Unit) alarm.</p> <p>OAIS, selects Optical Transport Unit (OTU) - Alarm Indication Signal which generates the polynomial numbers 11 (PN-11) over all OTU frame bits including FAS and MFAS continuously.</p> <p>OBDI, selects Optical Transport Unit (OTU) - Backward Defect Indication which generates a "1" for the BDI bit in the SM overhead field (byte 3, bit 5) continuously.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:TYPE**

LOF2, selects Loss of Frame which generates the errors in all the FAS bits continuously.

OOF1, selects Out of Frame which generates the errors in all the FAS bits for 5 consecutive OTU frames repetitively.

LOM, selects Loss of Multiframe which generates the errors in multiframe numbers for all the OTU frames continuously.

OOM, selects Out of Multiframe which generates the errors in multiframe numbers for 5 consecutive OTU frames repetitively.

OBlae, selects Optical Transport Unit (OTU) - Backward Incoming Alignment Error which generates a "1011" for the BEI/BIAE bits in the SM overhead field (byte 3, bits 1 to 4) continuously.

OAIe, selects Optical Transport Unit (OTU) - Incoming Alignment Error which generates a "1" for the IAE bit in the SM overhead field (byte 3, bit 6) continuously.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OTU1:TYPE OAIS
* SOUR:DATA:TEL:OTN:ALAR:OTU1:TYPE?
Returns OAIS

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:TYPE?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:TYPE?**

Description	<p>This query returns the type of Optical Transport Unit (OTU) alarm.</p> <p>At *RST, this value is set to OAIS.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:TYPE?</code>
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. 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>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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:TYPE?**

LOM, Loss of Multiframe (LOM) is selected as Optical Transport Unit (OTU) alarm.
OOM, Out of Multiframe (OOM) is selected as Optical Transport Unit (OTU) alarm.
OBIAE, Optical Transport Unit - Backward Incoming Alignment Error (OTU-BAIE) is selected as Optical Transport Unit (OTU) alarm.
OIAE, Optical Transport Unit - Incoming Alignment Error (OTU-AIE) is selected as Optical Transport Unit (OTU) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OTU1:TYPE OAIS
* SOUR:DATA:TEL:OTN:ALAR:OTU1:TYPE?
Returns OAIS

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]**

Description	<p>This command enables or disables the status of the Optical Transport Unit (OTU) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the OTU (Optical Transport Unit) alarm generation.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]**

Example(s) * SOUR:DATA:TEL:OTN:ALAR:OTU1:TYPE OAIS
 * SOUR:DATA:TEL:OTN:ALAR:OTU1 ON
 * SOUR:DATA:TEL:OTN:ALAR:OTU1? Returns 1

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE Transport Blazer and OTU3 is available
 for FTB/IQS-8140 Transport Blazer only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
 OTU[1..n]:TYPE
 * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
 OTU[1..n]?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]?**

Description	<p>This query returns the status of Optical Transport Unit (OTU) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of Optical Transport Unit (OTU) alarm generation.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]?**

Example(s) * SOUR:DATA:TEL:OTN:ALAR:OTU1:TYPE OAIS
 * SOUR:DATA:TEL:OTN:ALAR:OTU1 ON
 * SOUR:DATA:TEL:OTN:ALAR:OTU1? Returns 1

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
 OTU[1..n]:TYPE
 * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
 OTU[1..n]

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:MANual:TYPE**

Description	<p>This command selects the manual type of Optical Transport Unit (OTU) error.</p> <p>At *RST, this value is set to OBIP8.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:MANual:TYPE<wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the type of OTU error.</p> <p>OBIP8, selects the type of OTU-BIP-8 (ODU - Bit Interleave Parity-8) error.</p> <p>OBEI, selects the type of OTU-BEI (ODU - Backward Error Indication) error.</p> <p>FAS1, selects the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS, selects the type of MFAS (Multiframe Alignment Signal) error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:MANual:TYPE**

Example(s) * SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE?
Returns OBIP8

Note OTU1/OTU2 are available for FTB/QS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/QS-8140 Transport Blazer
only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:MANual:TYPE?

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:MANual:TYPE?**

Description	<p>This query returns the manual type of Optical Transport Unit (OTU) error.</p> <p>At *RST, this value is set to OBIP8.</p>
Syntax	<code>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:MANual:TYPE?</code>
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Optical Transport Unit (OTU) error.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (OUT-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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:MANual:TYPE?**

Example(s) * SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE?
Returns OBIP8

Note OTU1/OTU2 are available for FTB/QS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/QS-8140 Transport Blazer
only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AMOUNT**

Description	<p>This command sets the amount of Optical Transport Unit (OTU) error to be injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AMOUNT <wsp> <Amount> MAXimum MINimum</pre>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the amount of OTU (Optical Transport Unit) error. Choices are 1 through 50.</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AMOunt

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:OTU1:AMO 15* SOUR:DATA:TEL:OTN:ERR:OTU1:AMO? Returns 15
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AMOunt?

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:AMOUnt?**

Description	<p>This query returns the amount of Optical Transport Unit (OTU) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:AMOUnt? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<pre><Amount></pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AMOunt?

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Optical Transport Unit (OTU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:AMO?</p> <p>Returns 15</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NG/8130NGEv2 E Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOunt</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:INJect**

Description	<p>This command injects the type of Optical Transport Unit (OTU) error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:OTU1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:OTU1:INJ
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AMOUnt

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:AUTomated:TYPE

Description	<p>This command selects the type of Optical Transport Unit (OTU) error for automated injection.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:AUTomated:TYPE<wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the type of Optical Transport Unit (OTU) error for automated injection.</p> <p>OBIP8, selects the type of OTU-BIP-8 (ODU - Bit Interleave Parity-8) error.</p> <p>OBEI, selects the type of OTU-BEI (ODU - Backward Error Indication) error.</p> <p>FAS1, selects the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS, selects the type of MFAS (Multiframe Alignment Signal) error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:TYPE****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:TYPE?
Returns OBIP8

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:TYPE?
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:AUTomated:TYPE?**

Description	<p>This query returns the type of Optical Transport Unit (OTU) error for automated injection.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:AUTomated:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of Optical Transport Unit (OTU) error for the automated injection.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (OUT-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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:TYPE?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:TYPE?
Returns OBIP8

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:RATE**

Description	<p>This command sets the injection rate for the selected Optical Transport Unit (OTU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:RATE<wsp> <Rate> MAXimum MINimum</p>
Parameter(s)	<p>Rate: The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected Optical Transport Unit (OTU) error.</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:RATE****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:RATE?
Returns 1.0E-10

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:TYPE
* SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:RATE?
* SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Optical Transport Unit (OTU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:RATE?[<wsp> MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injection rate will be returned.</p>
Response Syntax	<p><Rate></p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:RATE?**

Response(s)	Rate: The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the injection rate for the selected Optical Transport Unit (OTU) error.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:RATE 1.0E-10 * SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:RATE? Returns 1.0E-10
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated

Description	<p>This command enables or disables the selected automated Optical Transport Unit (OTU) error at the rate specified or continuously.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated Optical Transport Unit (OTU) error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT ON
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT?
Returns 1

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated?**

Description	This query returns the status of automated Optical Transport Unit (OTU) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated Optical Transport Unit (OTU) error injection.

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT ON
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT?
Returns 1

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated

:SOURce[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:AUTomated:CONTInuous

Description	<p>This command enables or disables the continuous rate of automated Optical Transport Unit (OTU) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:AUTomated:CONTInuous<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated Optical Transport Unit (OTU) error injection continuously.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:CONTInuous****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:CONT
ON
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:CONT?
Returns 1
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT ON

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:CONTInuous?**

Description	<p>This query returns the status of continuous rate of automated Optical Transport Unit (OTU) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AUTomated:CONTInuous?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuous rate of automated Optical Transport Unit (OTU) error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:CONTInuous?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:CONT
ON
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT:CONT?
Returns 1
* SOUR:DATA:TEL:OTN:ERR:OTU1:AUT ON

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:AUTomated:CONTInuous

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE

Description	<p>This command selects the type of Optical Payload Unit (OPU) alarm.</p> <p>At *RST, this value is set to OMSim.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE<wsp>OMSim</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: OMSim.</p> <p>Selects the type of OPU alarm.</p> <p>OMSim, selects the alarm type as OPU-MSIM (Optical Payload Unit-Multiplex Structure Identifier Mismatch).</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:TYPE**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE? Returns OMSIM
Note	<ul style="list-style-type: none">* OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.* This command is available with ODU mux only.* FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:TYPE?**

Description	<p>This query returns the type of Optical Payload Unit (OPU) alarm.</p> <p>At *RST, this value is set to OMSim.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Alarm></p>
Response(s)	<p>Alarm: The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of OPU alarm. OMSIM, Multiplex Structure Identifier Optical Mismatch (OMSIM) is selected as Optical Payload Unit (OPU) alarm.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:TYPE?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE? Returns OMSIM
Note	<ul style="list-style-type: none">* OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.* This query is available with ODU mux only.* FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]

Description	<p>This command enables or disables the status of Optical Payload Unit (OPU) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]<wsp><Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the OPU (Optical Payload Unit) alarm generation.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1 ON* SOUR:DATA:TEL:OTN:ALAR:OPU1? Returns 1
Note	<ul style="list-style-type: none">* OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.* This command is available with ODU mux only.* FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]?

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

Description	This query returns the status of Optical Payload Unit (OPU) alarm generation. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Optical Payload Unit (OPU) alarm generation.

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

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1 ON* SOUR:DATA:TEL:OTN:ALAR:OPU1? Returns 1
Note	<ul style="list-style-type: none">* OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.* This query is available with ODU mux only.* FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TYPE

Description	<p>This command selects the type of Optical Data Unit (ODU) alarm.</p> <p>At *RST, this value is set to O AIS.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TYPE<wsp>O AIS OBDI OLCK OOCI OFSF OBSF OFS D OBS D LOFlom</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: O AIS OBDI OLCK OOCI OFSF OBSF OFS D OBS D LOFlom.</p> <p>Selects the type of ODU (Optical Data Unit) alarm.</p> <p>O AIS, selects 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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TYPE**

OBDI, selects 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).

OLCK, selects 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).

OOCI, selects ODU-BDI (ODU - Backward Defect Indication) which generates a "1" in the BDI (byte 3, bit 5) of the PM overhead field continuously.

OFSF, selects ODU-FSF (ODU - Forward Signal Fail) which generates a "00000001" pattern in the FTFL byte 0 continuously.

OBSF, selects ODU-BSF (ODU - Backward Signal Fail) which generates a "00000001" pattern in the FTFL byte 128 continuously.

OFSF, selects ODU-FSD (ODU - Forward Signal Degrade) which generates a "00000010" pattern in the FTFL byte 0 continuously.

OBSD, selects ODU-BSD (ODU - Backward Signal Degrade) which generates a "00000010" pattern in the FTFL byte 128 continuously.

LOFLom, ODU-LOFLOM (ODU-Loss of Frame Loss of Multiframe) which generate error continuously in FAS and MFAS of a multiplexed test case.

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TYPE

Example(s) * SOUR:DATA:TEL:OTN:ALAR:ODU1:TYPE OFSF
 * SOUR:DATA:TEL:OTN:ALAR:ODU1:TYPE?
 Returns OFSF

Note * OTU1/OTU2 are available for
 FTB/IQS-8120NGE/8130NGE/8130NGEv2
 Transport Blazer and OTU3 is available for
 FTB/IQS-8140 Transport Blazer only.
 * For **8120NGE/8130NGE/8130NGEv2** modules,
 choices are OAIS|OBDI|OLCK|OOCI|OFSF|
 OBSF|OFSO|OBSO|LOFlom.
 * For **8140** module, choices are OAIS|OBDI|
 OLCK|OOCI|OFSF|OBSF|OFSO| OBSO.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
 ODU[1..n]:TYPE?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TYPE?**

Description	<p>This query returns the type of Optical Data Unit (ODU) alarm.</p> <p>At *RST, this value is set to OAIS.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. 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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TYPE?**

OOCI, ODU - Open Connection Indication (ODU-OCI) is selected as Optical Data Unit (ODU) alarm.
OFSF, ODU - Forward Signal Fail (ODU-FSF) is selected as Optical Data Unit (ODU) alarm.
OBSF, ODU - Backward Signal Fail (ODU-BSF) is selected as Optical Data Unit (ODU) alarm.
OFSD, ODU - Forward Signal Degrade (ODU-FSD) is selected as Optical Data Unit (ODU) alarm.
OBSD, ODU - Backward Signal Degrade (ODU-BSD) is selected as Optical Data Unit (ODU) alarm.
LOFLOM, ODU Loss of Frame Loss of Multiframe (ODU-LOFLOM) is selected as Optical Data Unit (ODU) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:ODU1:TYPE OFSF
* SOUR:DATA:TEL:OTN:ALAR:ODU1:TYPE?
Returns OFSF

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]**

Description	<p>This command enables or disables the status of Optical Data Unit (ODU) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n] <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the ODU (Optical Data Unit) alarm generation.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]**

Example(s) * SOUR:DATA:TEL:OTN:ALAR:ODU1:TYPE OFSF
 * SOUR:DATA:TEL:OTN:ALAR:ODU1 ON
 * SOUR:DATA:TEL:OTN:ALAR:ODU1? Returns 1

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
 8130NGE/8130NGEv2 Transport Blazer and OTU3
 is available for FTB/IQS-8140 Transport Blazer
 only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
 ODU[1..n]:TYPE
 * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
 ODU[1..n]?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]?**

Description	<p>This query returns the status of Optical Data Unit (ODU) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 RESPONSE DATA> element.</p> <p>Returns the status of Optical Data Unit (ODU) alarm generation.</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]?

Example(s)

- * SOUR:DATA:TEL:OTN:ALAR:ODU1:TYPE OFSF
- * SOUR:DATA:TEL:OTN:ALAR:ODU1 ON
- * SOUR:DATA:TEL:OTN:ALAR:ODU1? Returns 1

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

- * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TYPE
- * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]

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

Description	<p>This command selects the manual type of Optical Data Unit (ODU) error.</p> <p>At *RST, this value is set to OBIP8.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:MANual:TYPE<wsp>OBIP8 OBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the type of ODU (Optical Data Unit) error. OBIP8, selects 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, selects 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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:MANual:TYPE**

Example(s) * SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE?
Returns OBIP8

Note OTU1/OTU2 are available for FTB/QS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/QS-8140 Transport Blazer
only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:MANual:TYPE?

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

Description	<p>This query returns the manual type of Optical Data Unit (ODU) error.</p> <p>At *RST, this value is set to OBIP8.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:MANual:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Optical Channel Data Unit (ODU) error.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as Optical Channel Data Unit (ODU) error.</p> <p>OBEI, ODU - Backward Error Indication (ODU-BEI) is selected as Optical Data Unit (ODU) error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:MANual:TYPE?**

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE?
Returns OBIP8

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:MANual:TYPE

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:MANual:TCM[1..n]:TYPE**

Description This command selects the manual type of Optical Data Unit (ODU) error for the TCM level.

At *RST, this value is set to TBIP8.

Syntax :SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:MANual:TCM[1..n]:TYPE<wsp>
TBIP8|TBEI

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:MANual:TCM[1..n]:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the manual type of Optical Data Unit (ODU) error.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TCM1:TYPE TBIP8* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TCM1:TYPE? Returns TBIP8
See Also	<ul style="list-style-type: none">* 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?**

Description	This query returns the manual type of Optical Data Unit (ODU) error for the TCM level. At *RST, this value is set to TBIP8.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:MANual:TCM[1..n]:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:MANual:TCM[1..n]:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Optical Data Unit (ODU) error.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TCM1:TYPE? Returns TBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]TYPE</p>

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:TCM[1..n]:AMOut****Description**

This command sets the amount of Optical Data Unit (ODU) error to inject for the TCM level.

At *RST, this value is set to 1.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:  
ODU[1..n]:TCM[1..n]:AMOut <wsp>  
<Amount> | MAXimum | MINimum
```

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:TCM[1..n]:AMOUNT**

Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the amount of Optical Data Unit (ODU) error to inject.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:TCM[1..n]:AMOUNT?</p>

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:TCM[1..n]:AMOUnt?**

Description	<p>This query returns the amount of Optical Channel Data Unit (ODU) error to inject for the TCM level.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:AMOUnt?<wsp> <Amount> MAXimum MINimum</pre>
Parameter(s)	<p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount will be returned.</p>
Response Syntax	<pre><Amount></pre>

SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:AMOut?

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of Optical Data Unit (ODU) error.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TCM1:TYPE TBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AMO? Returns 15
See Also	* 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]:TCM[1..n]:AMOut

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:MANual:TCM[1..n]:TYPE**

Description	<p>This command selects the manual type of Optical Data Unit (ODU) error for ODU1e/2e.</p> <p>At *RST, this value is set to OBIP8.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:MANual:TCM[1..n]:TYPE<wsp> TBIP8 TBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI.</p> <p>Selects the manual type of Optical Data Unit (ODU) error for ODU1e/2e.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:MANual:TCM[1..n]:TYPE

Example(s) * SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TCM1:
TYPE TBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TCM1:
TYPE? Returns TBIP8

Note OTU1/OTU2 are available for FTB/QS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/QS-8140 Transport Blazer
only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:MANual:TCM[1..n]TYPE?

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:MANual:TCM[1..n]:TYPE?**

Description	This query returns the manual type of Optical Data Unit (ODU) error for ODU1e/2e. At *RST, this value is set to OBIP8.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:MANual:TCM[1..n]:TYPE?
Parameter(s)	None
Response Syntax	<Error>

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:MANual:TCM[1..n]:TYPE?

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Optical Data Unit (ODU) error.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TCM1:TYPE? Returns TBIP8</p>
See Also	<p>* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:ODU[1..n]:E:MANual:TCM[1..n]TYPE</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:TCM[1..n]:AMOUNT****Description**

This command sets the amount of Optical Data Unit (ODU) error to inject for ODU1e/2e of the TCM level.

At *RST, this value is set to 1.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:TCM[1..n]:AMOUNT <wsp>
<Amount> | MAXimum | MINimum

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:TCM[1..n]:AMOut**

Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Sets the amount of ODU (Optical Data Unit) error. Choices are 1 through 50. The default setting is 1.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:TCM1:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:TCM1:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E:MANual:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E:TCM[1..n]:AMOut?</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:TCM[1..n]:AMOUNT?**

Description	<p>This query returns the amount of Optical Channel Data Unit (ODU) error to inject for ODU1e/2e of the TCM level.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:TCM[1..n]:AMOUNT?<wsp> <Amount> MAXimum MINimum</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount will be returned.</p>
Response Syntax	<pre><Amount></pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:AMOunt?

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Optical Data Unit (ODU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:TCM1:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:TCM1:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E:MANual:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E:TCM[1..n]:AMOunt</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AMOUnt?**

Description	<p>This query returns the amount of Optical Data Unit (ODU) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AMOUnt? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<pre><Amount></pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AMOut?

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Optical Data Unit (ODU) error.</p>
Example(s)	<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>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<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]:AMOut</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AMOUnt?**

Description	<p>This query returns the amount of Optical Data Unit (ODU) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AMOUnt? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<pre><Amount></pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AMOut?

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Optical Data Unit (ODU) error.</p>
Example(s)	<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>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<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]:AMOut</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:INJect**

Description	<p>This command injects the type of Optical Data Unit (ODU) error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:ODU1:INJ
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AMOUNT

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:INJect

Description	<p>This command injects the type of Optical Data Unit (ODU) error for the TCM level.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:INJect</pre>
Parameter(s)	None
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TCM1: TYPE TBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:INJ</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:TCM[1..n]:INJect**

Description	<p>This query returns the amount of Optical Channel Data Unit (ODU) error to inject for ODU1e/2e of the TCM level.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TCM1: TYPE TBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:E:TCM1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:ODU1:E:TCM1:INJ

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TYPE

Description	<p>This command selects the type of Optical Data Unit (ODU) error for automated injection.</p> <p>At *RST, this value is set to OBIP8.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TYPE<wsp>OBIP8 OBEI</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the type of Optical Data Unit (ODU) error for automated injection.</p> <p>OBIP8, selects 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, selects 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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:TYPE****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE?
Returns OBIP8

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TYPE?

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:TYPE?**

Description	<p>This query returns the type of Optical Data Unit (ODU) error for automated injection.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:AUTomated:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>
Response(s)	<p>Error: The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Optical Data Unit (ODU) error for the automated injection. OBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as Optical Data Unit (ODU) error. OBEI, ODU - Backward Error Indication (ODU-BEI) is selected as Optical Data Unit (ODU) error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:TYPE?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE?
Returns OBIP8

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:RATE

Description	<p>This command sets the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:RATE<wsp><Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected Optical Data Unit (ODU) error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:RATE****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE?
Returns 1.0E-10

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:RATE?
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:RATE?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Rate></p>
Response(s)	<p>Rate: The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the injection rate for the selected Optical Data Unit (ODU) error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:RATE?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE?
Returns 1.0E-10

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated

Description This command enables or disables the selected automated Optical Data Unit (ODU) error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated <wsp> <Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated Optical Data Unit (ODU) error injection.

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT ON
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT?
Returns 1

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated?**

Description	This query returns the status of automated Optical Data Unit (ODU) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:AUTomated?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated Optical Data Unit (ODU) error injection.

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:RATE
1.0E-10
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT ON
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT?
Returns 1

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:CONTInuous**

Description This command enables or disables the continuous rate of automated Optical Data Unit (ODU) error injection.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:CONTInuous<wsp>
<Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the continuous rate of automated Optical Data Unit (ODU) error injection.

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:CONTInuous****Example(s)**

* 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
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT ON

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:CONTInuous?

:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:CONTInuous?

Description	<p>This query returns the status of continuous rate of automated Optical Data Unit (ODU) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:CONTInuous?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuous rate of automated Optical Data Unit (ODU) error injection.</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:CONTInuous?****Example(s)**

* 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
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT ON

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU1:AUTomated:TYPE
* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU1:AUTomated
* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU1:AUTomated:CONTInuous

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:FEC:MANual:TYPE

Description	<p>This command selects the manual type of Forward Error Correction (FEC) error.</p> <p>At *RST, this value is set to FCCW.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:FEC:MANual:TYPE<wsp>FCCW FUCW FCSYmb FCBit FSCW</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit FSCW.</p> <p>Selects the type of FEC (Forward Error Correction) error.</p> <p>FCCW, selects the FCCORR-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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:FEC:MANual:TYPE**

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.

FSCW, selects the FEC-STRESS-CW (Forward Error Correction - Stress - Codeword) which generates correctable errors composed of a random number of symbol errors (less or equal to 8) containing a random number of bits distributed all over the OTU frame.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN:TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN:TYPE? Returns FCCW

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERROR:OTU[1..n]:FEC:MANual:TYPE?

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:FEC:MANual:TYPE?**

Description	<p>This query returns the manual type Forward Error Correction (FEC) error.</p> <p>At *RST, this value is set to FCCW.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:FEC:MANual:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Forward Error Correction (FEC) error.</p> <p>FCCW, Forward Error Correction - Correctable - Codeword (FEC-CORR-CW) is selected as Forward Error Correction (FEC) error.</p> <p>FUCW, Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) is selected as Forward Error Correction (FEC) error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:MANual:TYPE?**

FCSYMB, Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) is selected as Forward Error Correction (FEC) error.

FCBIT, Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) is selected as Forward Error Correction (FEC) error.

FSCW, Forward Error Correction - Stress - Codeword (FEC-STRESS-CW) is selected as Forward Error Correction (FEC) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN:TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN:TYPE? Returns FCCW

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:FEC:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AMOut**

Description This command sets the amount of Forward Error Correction (FEC) error to be injected into the instrument.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AMOut <wsp> <Amount>
|MAXimum|MINimum

Parameter(s) Amount:
The program data syntax for the parameter is defined as a <numeric_value> element.
The allowed <numeric_value> elements for this parameter are: MAXimum|MINimum.
MAXimum allows to set the instrument to the greatest supported value.
MINimum allows to set the instrument to the smallest supported value.
Sets the amount of FEC (Forward Error Correction) error.
Choices are 1 through 50.

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AMOut**

Example(s)	* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN: TYPE FCCW * SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AMO 15 * SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AMO? Returns 15
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AMOut?

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AMOut?

Description	<p>This query returns the amount of Forward Error Correction (FEC) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AMOut?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for <Amount> is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<pre><Amount></pre>

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

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Forward Error Correction (FEC) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN:TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AMO?</p> <p>Returns 15</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<p>* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:OTU[1..n]:FEC:MANual:TYPE</p> <p>* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:OTU[1..n]:FEC:AMOUNT</p>

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

Description	<p>This command injects the type of Forward Error Correction (FEC) error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:FEC:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN: TYPE FCCW * SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AMO 15 * SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:INJ
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:FEC:MANual:TYPE * SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:FEC:AMOUNT

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:FEC:AUTomated:TYPE**

Description	<p>This command selects the type of Forward Error Correction (FEC) error for automated injection.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:FEC:AUTomated:TYPE<wsp>FCCW FUCW FCSYmb FCBit FSCW</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit FSCW.</p> <p>Selects the type of Forward Error Correction (FEC) error for automated injection.</p> <p>FCCW, selects the FCCORR-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>

:SOURce[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:FEC:AUTomated:TYPE

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 1bit in error.

FSCW, selects the FEC-STRESS-CW (Forward Error Correction - Stress - Codeword) which generates correctable errors composed of a random number of symbol errors (less or equal to 8) containing a random number of bits distributed all over the OTU frame.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:TYPE? Returns FCCW

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERROR:OTU[1..n]:FEC:AUTomated:TYPE?
* SOURce[1..n]:DATA:TELEcom:OTN:ERROR:OTU[1..n]:FEC:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERROR:OTU[1..n]:FEC:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:FEC:AUTomated:TYPE?**

Description	This query returns the type of Forward Error Correction (FEC) error for automated injection. At *RST, this value is set to BERRor.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:FEC:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:FEC:AUTomated:TYPE?**

Response(s)

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the type of Forward Error Correction (FEC) error for the automated injection.

FCCW, Forward Error Correction - Correctable - Codeword (FEC-CORR-CW) is selected as Forward Error Correction (FEC) error.

FUCW, Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) is selected as Forward Error Correction (FEC) error.

FCSYMB, Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) is selected as Forward Error Correction (FEC) error.

FCBIT, Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) is selected as Forward Error Correction (FEC) error.

FSCW, Forward Error Correction - Stress - Codeword (FEC-STRESS-CW) is selected as Forward Error Correction (FEC) error.

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:TYPE?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
TYPE? Returns FCCW

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:RATE**

Description	<p>This command sets the injection rate for the selected Forward Error Correction (FEC) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated:RATE<wsp> <Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected Forward Error Correction (FEC) error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:RATE****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
RATE 1.0E-10
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
RATE? Returns 1.0E-10

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:RATE?
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Forward Error Correction (FEC) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated:RATE? [<wsp> MAXimum MINimum]</pre>
Parameter(s)	None
Response Syntax	<Rate>
Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Forward Error Correction (FEC) error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:RATE?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
RATE 1.0E-10
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
RATE? Returns 1.0E-10

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated

Description	<p>This command enables or disables the selected automated Forward Error Correction (FEC) error at the rate specified or continuously.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated Forward Error Correction (FEC) error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
RATE 1.0E-10
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT ON
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT?
Returns 1

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated?

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated?

Description	<p>This query returns the status of automated Forward Error Correction (FEC) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated Forward Error Correction (FEC) error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
RATE 1.0E-10
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT ON
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT?
Returns 1

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:FEC:AUTomated:CONTInuous**

Description	<p>This command enables or disables the continuous rate of automated Forward Error Correction (FEC) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:FEC:AUTomated:CONTInuous<wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the continuous rate of automated Forward Error Correction (FEC) error injection.</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:CONTInuous****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
CONT ON
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
CONT? Returns 1
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT ON

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:TYPE
* SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated
* SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:CONTInuous?**

Description	<p>This query returns the status of continuous rate of automated Forward Error Correction (FEC) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:AUTomated:CONTInuous?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuous rate of automated Forward Error Correction (FEC) error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:CONTInuous?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
CONT ON
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT:
CONT? Returns 1
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AUT ON

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AUTomated:CONTInuous

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:TYPE**

Description	<p>This command selects the type of Optical Payload Unit (OPU) alarm.</p> <p>At *RST, this value is set to OMSim.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE<wsp>OMSim</p>
Parameter(s)	<p>Alarm: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. Selects the type of OPU (Optical Payload Unit) alarm. OMSim, selects the alarm type as OPU-MSIM (Optical Payload Unit-Multiplex Structure Identifier Mismatch).</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS * SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE? Returns OMSIM</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:TYPE?**

Description	<p>This query returns the type of Optical Payload Unit (OPU) alarm.</p> <p>At *RST, this value is set to OMSim.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE? <wsp>OMSim</code>
Parameter(s)	None
Response Syntax	<code><Alarm></code>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <code><Alarm></code> is defined as a <code><CHARACTER RESPONSE DATA></code> element. Returns the type of Optical Payload Unit (OPU) alarm.</p> <p>OMSIM, OMSIM is selected as Optical Payload Unit (OPU) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS * SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE? Returns OMSIM</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]**

Description	<p>This command enables or disables the status of Optical Payload Unit (OPU) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]<wsp><Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. Enables or disables the OPU (Optical Payload Unit) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1 ON* SOUR:DATA:TEL:OTN:ALAR:OPU1? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]?

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

Description	<p>This query returns the type of Optical Payload Unit (OPU) alarm.</p> <p>At *RST, this value is set to OMSim.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]?<wsp>OMSim</code>
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of Optical Payload Unit (OPU) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1 ON* SOUR:DATA:TEL:OTN:ALAR:OPU1? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TCM[1..n]:TYPE

Description This command allows the selection of alarm type for TCM level of Optical Data Unit (ODU) alarm.

At *RST, this value is set to TCMi-LTC.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:TYPE<wsp>TLTC|TBDI|
TIQE|TBIAE

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TCM[1..n]:TYPE

Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TLTC TBDI TIQE TBIAE</p> <p>Selects the type of ODU (Optical Data Unit) alarm.</p> <p>TLTC, selects TCMi - Loss of Tandem Connection as the type of ODU (Optical Data Unit) alarm.</p> <p>TBDI, selects TCMi - Backward Defect Indication as the type of ODU (Optical Data Unit) alarm.</p> <p>TIQE, selects TCMi - Incoming Alignment Error as the type of ODU (Optical Data Unit) alarm.</p> <p>TBIAE, selects TCMi - Backward Incoming Alignment Error as the type of ODU (Optical Data Unit) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:TCM1:TYPE TLTC</p> <p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:TCM1:TYPE? Returns TLTC</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TCM[1..n]:TYPE?

Description	This query returns the selected alarm type for TCM level of Optical Data Unit (ODU) alarm. At *RST, this value is set to TCMi-LTC.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TCM[1..n]:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:TYPE?****Response(s)****Alarm:**

The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the selected alarm type for TCM level of Optical Data Unit (ODU) alarm.

TLTC, TCMi - Loss of Tandem Connection is selected as the type of ODU (Optical Data Unit) alarm.

TBDI, TCMi - Backward Defect Indication is selected as the type of ODU (Optical Data Unit) alarm.

TIQE, TCMi - Incoming Alignment Error is selected as the type of ODU (Optical Data Unit) alarm.

TBIAE, TCMi - Backward Incoming Alignment Error is selected as the type of ODU (Optical Data Unit) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:ODU1:TCM1:TYPE
TLTC

* SOUR:DATA:TEL:OTN:ALAR:ODU1:TCM1:
TYPE? Returns TLTC

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:TYPE?

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODUTCM[1..n]

Description This command enables or disables the selected alarm type.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODUTCM[1..n] <wsp> <Level>, <Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODUTCM[1..n]**

Parameter(s)	<p>Level: The program data syntax for the first parameter is defined as a <NRI NUMERIC PROGRAM DATA> element. Selects the TCM level for alarm type.</p> <p>Set: The program data syntax for the second parameter is defined as a <BOOLEAN PROGRAM DATA> element. Enables or disables the selected alarm type.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:ODU1:TCM1:TYPE TLTC* SOUR:DATA:TEL:OTN:ALAR:ODUTCM2 1, ON
See Also	<ul style="list-style-type: none">* 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]?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODUTCM[1..n]?**

Description	<p>This query returns the status of the selected alarm type.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODUTCM[1..n]?<wsp><Level></p>
Parameter(s)	<p>Level: The program data syntax for the first parameter is defined as a <NRI NUMERIC PROGRAM DATA> element. Selects the TCM level for alarm type.</p>
Response Syntax	<p><Set></p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODUTCM[1..n]?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the selected alarm type.
Example(s)	* SOUR:DATA:TEL:OTN:ALAR:ODUTCM2 1,ON * SOUR:DATA:TEL:OTN:ALAR:ODUTCM2? 1 Returns 1
See Also	* 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]

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TCM[1..n]:TYPE**

Description

This command selects the type of Optical Data Unit (ODU) alarm for ODU1e/2e of the TCM level.

At *RST, this value is set to OAI5.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TCM[1..n]:TYPE<wsp>TLTC|TBDI|
TIQE|TBIAE

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:TYPE

Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TLTC TBDI TIQE TBIAE.</p> <p>Selects the type of ODU (Optical Data Unit) alarm.</p> <p>TLTC, selects TCMi - Loss of Tandem Connection as the type of ODU (Optical Data Unit) alarm.</p> <p>TBDI, selects TCMi - Backward Defect Indication as the type of ODU (Optical Data Unit) alarm.</p> <p>TIQE, selects TCMi - Incoming Alignment Error as the type of ODU (Optical Data Unit) alarm.</p> <p>TBIAE, selects TCMi - Backward Incoming Alignment Error as the type of ODU (Optical Data Unit) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TCM1:TYPE TLTC</p> <p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TCM1:TYPE? Returns TLTC</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TCM[1..n]</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TCM[1..n]:TYPE?**

Description	This query returns the type of Optical Data Unit (ODU) alarm for ODU1e/2e of the TCM level. At *RST, this value is set to OAIS.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARM:
ODU[1..n]:E:TCM[1..n]:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Optical Data Unit (ODU) alarm.</p> <p>TLTC, TCMi - Loss of Tandem Connection is selected as the type of ODU (Optical Data Unit) alarm.</p> <p>TBDI, TCMi - Backward Defect Indication is selected as the type of ODU (Optical Data Unit) alarm.</p> <p>TIAE, TCMi - Incoming Alignment Error is selected as the type of ODU (Optical Data Unit) alarm.</p> <p>TBIAE, TCMi - Backward Incoming Alignment Error is selected as the type of ODU (Optical Data Unit) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TCM1:TYPE? Returns TBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:E:MANual:TCM[1..n]TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
E:ODUTCM[1..n]**

Description This command enables or disables the status of Optical Channel Data Unit (ODU) alarm generation for ODU1e/2e.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
E:ODUTCM[1..n]<wsp><Level>,<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
E:ODUTCM[1..n]**

Parameter(s)	<p>Level: The program data syntax for the first parameter is defined as a <NRI NUMERIC PROGRAM DATA> element. Selects the TCM level for alarm type.</p> <p>Set: The program data syntax for the second parameter is defined as a <Boolean Program Data> element. Enables or disables the ODU (Optical Data Unit) alarm generation.</p>
Example(s)	* SOUR:DATA:TEL:OTN:ALAR:E:ODUTCM1 1, ON
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]?

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:E:ODUTCM[1..n]?

Description	<p>This query returns the status of Optical Data Unit (ODU) alarm generation for ODU1e/2e.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:E:ODUTCM[1..n]?<wsp><Level></p>
Parameter(s)	<p>Level: The program data syntax for the first parameter is defined as a <NRI NUMERIC PROGRAM DATA> element. Selects the TCM level for alarm type.</p>
Response Syntax	<p><Set></p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
E:ODUTCM[1..n]?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Optical Data Unit (ODU) alarm generation.
Example(s)	* SOUR:DATA:TEL:OTN:ALAR:E:ODUTCM2 1,ON * SOUR:DATA:TEL:OTN:ALAR: E:ODUTCM2? 1 Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]?

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:HISTory?

Description	<p>This query returns the history status of Optical Transport Unit (OTU) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:HISTory? <wsp>LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBlae</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBlae.</p> <p>Selects the type of OTU (Optical Transport Unit) alarm.</p> <p>LOF2, selects LOF (Loss of Frame) when OOF is present for at least 3 ms.</p> <p>OOF1, selects OOF (Out of Frame) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU frames.</p> <p>LOM, selects LOM (Loss of Multiframe) when OOM is present for at least 3 ms.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:HISTory?**

OOM, selects OOM (Out of Multiframe) when MFAS are in error for at least 5 consecutive OTU frames.

OAIS, selects OTU-AIS (OTU - Alarm Indication Signal) when polynomial number 11 (PN-11) is over all OTU frame bits including FAS and MFAS (Multiframe Alignment Signal) for at least 3 consecutive 8192 bit-interval.

OTIM, selects OTU-TIM (OTU - Trace Identifier Mismatch) when expected SM SAPI and/or SM DAPI do not match the received SM SAPI (Source Access Point Identifier) and/or DAPI (Destination Access Point Identifier) for at least 3 consecutive TTI of the 256 frames multiframe.

OBDI, selects OTU-BDI (OTU - Backward Defect Indication) when the BDI bit in the SM overhead field (byte 3, bit 5) is "1" for at least 5 consecutive OTU frames.

OIAE, selects 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.

OBlae, selects 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.

Response Syntax <History>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Optical Transport Unit (OTU) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OTU1:TYPE OAIS* SOUR:DATA:TEL:OTN:ALAR:OTU1 ON* FETC:DATA:TEL:OTN:ALAR:OTU1:HIST? OAIS <p>Returns the alarm history status.</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]?

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:SEConds?

Description	<p>This query returns the number of seconds within which Optical Transport Unit (OTU) alarm occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:SEConds? <wsp>LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBIAe</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBIAe.</pre> <p>Selects the type of OTU (Optical Transport Unit) alarm.</p> <p>LOF2, selects LOF (Loss of Frame) when OOF is present for at least 3 ms.</p> <p>OOF1, selects OOF (Out of Frame) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU frames.</p> <p>LOM, selects LOM (Loss of Multiframe) when OOM is present for at least 3 ms.</p>

:FETCh[1..n]:DATA:TELecom:OTN:ALARm: OTU[1..n]:SEConds?

OOM, selects OOM (Out of Multiframe) when MFAS are in error for at least 5 consecutive OTU frames.

OAIS, selects OTU-AIS (OTU - Alarm Indication Signal) when polynomial number 11 (PN-11) is over all OTU frame bits including FAS and MFAS for at least 3 consecutive 8192 bit-interval.

OTIM, selects 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 (Destination Access Point Identifier) for at least 3 consecutive TTI (Trail Trace Identifier) of the 256 frames multiframe.

OBDI, selects OTU-BDI (OTU - Backward Defect Indication) when the BDI bit in the SM overhead field (byte 3, bit 5) is "1" for at least 5 consecutive OTU frames.

OIAE, selects 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.

OBlae, selects OTU-BIAE (OTU - Backward Incoming Alignment Error) when BEI/BIAE bits in the SM overhead field (byte 3, bits 1 to 4) are "1011" for at least 3 consecutive frames.

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Transport Unit (OTU) alarm.
Example(s)	* SOUR:DATA:TEL:OTN:ALAR:OTU1:TYPE OAIS * SOUR:DATA:TEL:OTN:ALAR:OTU1 ON * FETC:DATA:TEL:OTN:ALAR:OTU1:SEC? OAIS Returns the number of seconds of OTU alarm.
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]?

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:CURRent?

Description	<p>This query returns the current status of Optical Transport Unit (OTU) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:CURRent? <wsp>LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBlae</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBlae.</p> <p>Selects the type of OTU (Optical Transport Unit) alarm.</p> <p>LOF2, selects LOF (Loss of Frame) when OOF is present for at least 3 ms.</p> <p>OOF1, selects OOF (Out of Frame) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU frames.</p> <p>LOM, selects LOM (Loss of Multiframe) when OOM is present for at least 3 ms.</p>

**:FETCh[1..n]:DATA:TELecom:OTN:ALARm:
OTU[1..n]:CURRent?**

OOM, selects OOM (Out of Multiframe) when MFAS are in error for at least 5 consecutive OTU frames.

OAIS, selects OTU-AIS (OTU - Alarm Indication Signal) when polynomial number 11 (PN-11) is over all OTU frame bits including FAS and MFAS for at least 3 consecutive 8192 bit-interval.

OTIM, selects OTU-TIM (OTU - Trace Identifier Mismatch) when expected SM SAPI (Source Access Point Identifier) and/or SM DAPI do not match the received SM SAPI and/or DAPI (Destination Access Point Identifier) for at least 3 consecutive TTI (Trail Trace Identifier) of the 256 frames multiframe.

OBDI, selects OTU-BDI (OTU - Backward Defect Indication) when the BDI bit in the SM overhead field (byte 3, bit 5) is "1" for at least 5 consecutive OTU frames.

OIAE, selects 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.

OBlae, selects OTU-BIAE (OTU - Backward Incoming Alignment Error) when BEI/BIAE bits in the SM overhead field (byte 3, bits 1 to 4) are "1011" for at least 3 consecutive frames.

Response Syntax <Current>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of Optical Transport Unit (OTU) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OTU1:TYPE OAIS* SOUR:DATA:TEL:OTN:ALAR:OTU1 ON* FETC:DATA:TEL:OTN:ALAR:OTU1:CURR? OAIS <p>Returns the current alarm status.</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]?

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:HISTory?

Description	<p>This query returns the history status of Optical Transport Unit (OTU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:HISTory? <wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the type of OTU (Optical Transport Unit) error.</p> <p>OBIP8, selects the type of OTU-BIP-8 (ODU - Bit Interleave Parity-8) error.</p> <p>OBEI, selects the type of OTU-BEI (ODU - Backward Error Indication) error.</p> <p>FAS1, selects the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS, selects the type of MFAS (Multiframe Alignment Signal) error.</p>
Response Syntax	<History>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:OTU1:AMO 15* SOUR:DATA:TEL:OTN:ERR:OTU1:INJ* FETC:DATA:TEL:OTN:ERR:OTU1:HIST? OBIP8 <p>Returns the error history status.</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:AMOUNT* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:INJECT

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:SECOnds?**

Description	<p>This query returns the number of seconds within which Optical Transport Unit (OTU) error occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:SECOnds?<wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the type of OTU (Optical Transport Unit) error.</p> <p>OBIP8, selects the type of OTU-BIP-8 (ODU - Bit Interleave Parity-8) error.</p> <p>OBEI, selects the type of OTU-BEI (ODU - Backward Error Indication) error.</p> <p>FAS1, selects the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS, selects the type of MFAS (Multiframe Alignment Signal) error.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:SEConds?**

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Optical Transport Unit (OTU) error.

Example(s)

- * SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE OBIP8
- * SOUR:DATA:TEL:OTN:ERR:OTU1:AMO 15
- * SOUR:DATA:TEL:OTN:ERR:OTU1:INJ
- * FETC:DATA:TEL:OTN:ERR:OTU1:SEC? OBIP8

Returns the number of errored seconds.

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

- * SOURce[1..n]:DATA:TELEcom:OTN:ERROr:OTU[1..n]:TYPE
- * SOURce[1..n]:DATA:TELEcom:OTN:ERROr:OTU[1..n]:AMOUnt
- * SOURce[1..n]:DATA:TELEcom:OTN:ERROr:OTU[1..n]:INJect

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:CURREnt?

Description	<p>This query returns the current status of Optical Transport Unit (OTU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:CURREnt? <wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the type of OTU (Optical Transport Unit) error.</p> <p>OBIP8, selects the type of OTU-BIP-8 (ODU - Bit Interleave Parity-8) error.</p> <p>OBEI, selects the type of OTU-BEI (ODU - Backward Error Indication) error.</p> <p>FAS1, selects the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS, selects the type of MFAS (Multiframe Alignment Signal) error.</p>
Response Syntax	<Current>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:CURREnt?**

Response(s) Current:
The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the current status of Optical Transport Unit (OTU) 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:OTN:ERR:OTU1:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:AMO 15
* SOUR:DATA:TEL:OTN:ERR:OTU1:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:CURR? OBIP8
Returns the current error status.

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:INJECT

:FETCh[1..n]:DATA:TELEcom:OTN:ERror: OTU[1..n]:COUNT?

Description	<p>This query returns the count of Optical Transport Unit (OTU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:ERror: OTU[1..n]:COUNT?<wsp>OBIP8 OBEI FAS1 MFAS
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the type of OTU (Optical Transport Unit) error.</p> <p>OBIP8, selects the type of OTU-BIP-8 (ODU - Bit Interleave Parity-8) error.</p> <p>OBEI, selects the type of OTU-BEI (ODU - Backward Error Indication) error.</p> <p>FAS1, selects the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS, selects the type of MFAS (Multiframe Alignment Signal) error.</p>
Response Syntax	<Count>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:COUNT?

Response(s)	Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of Optical Transport Unit (OTU) error.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:OTU1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:OTU1:INJ * FETC:DATA:TEL:OTN:ERR:OTU1:COUN? OBIP8 Returns the error count.
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AMOUNT * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:INJECT

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:RATE?

Description	<p>This query returns the current rate of Optical Transport Unit (OTU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:RATE? <wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the type of OTU (Optical Transport Unit) error.</p> <p>OBIP8, selects the type of OTU-BIP-8 (ODU - Bit Interleave Parity-8) error.</p> <p>OBEI, selects the type of OTU-BEI (ODU - Backward Error Indication) error.</p> <p>FAS1, selects the type of FAS (Frame Alignment Signal) error.</p> <p>MFAS, selects the type of MFAS (Multiframe Alignment Signal) error.</p>
Response Syntax	<Rate>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:RATE?**

Response(s)	Rate: The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the current rate of Optical Transport Unit (OTU) error.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:OTU1:MAN:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:OTU1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:OTU1:INJ * FETC:DATA:TEL:OTN:ERR:OTU1:RATE? OBIP8 Returns the error rate.
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:AMOUNT * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:INJECT

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:HISTory?**

Description	<p>This query returns the history status of Optical Data Unit (ODU) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU [1..n]:HISTory?<wsp>OAIS OBDI OLCK OOCI OFSF OBSF OTIM OFSd OBSd LOFLom</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>

:FETCH[1..n]:DATA:TELEcom:OTN:ALARM: ODU[1..n]:HISTory?

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 - Backward Defect indication) 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.

OFSF, ODU-FSF (ODU - Forward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000001".

OBSF, ODU-BSF (ODU - Backward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000001".

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.

OFSF, 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".

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:HISTory?**

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.

Response Syntax <History>

Response(s) History:
The response data syntax for <History> 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.

:FETCh[1..n]:DATA:TELecom:OTN:ALARm: ODU[1..n]:HISTory?

Example(s)

- * SOUR:DATA:TEL:OTN:ALAR:ODU1:TYPE OAIS
- * SOUR:DATA:TEL:OTN:ALAR:ODU1 ON
- * FETC:DATA:TEL:OTN:ALAR:ODU1:HIST? OAIS

Returns the alarm history status.

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

- * SOURce[1..n]:DATA:TELecom:OTN:ALARm:
ODU[1..n]:TYPE
- * SOURce[1..n]:DATA:TELecom:OTN:ALARm:
ODU[1..n]

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:SEConds?

Description	<p>This query returns the number of seconds within which Optical Data Unit (ODU) alarm occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU [1..n]:SEConds?<wsp>OAIS OBDI OLCK OOCl OFSF OBSF OTIM OFSD OBSd LOFLom</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OAIS OBDI OLCK OOCl 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>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:SEConds?

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 - Backward Defect indication) 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.

OFSF, ODU-FSF (ODU - Forward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000001".

OBSF, ODU-BSF (ODU - Backward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000001".

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 math the expected SAPI and/or DAPI. This alarm is only available when TIM SAPI or DAPI is enabled.

OFSF, 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".

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:SEConds?

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.

Response Syntax	<Seconds>
Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Data Unit (ODU) alarm.
Example(s)	* SOUR:DATA:TEL:OTN:ALAR:ODU1:TYPE OAIS * SOUR:DATA:TEL:OTN:ALAR:ODU1 ON * FETC:DATA:TEL:OTN:ALAR:ODU1:SEC? OAIS Returns the number of seconds of ODU alarm.
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CURRent?

Description	<p>This query returns the current status of Optical Data Unit (ODU) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:CURRent?<wsp>OAIS OBDI OLCK OOCl OFSF OBSF OTIM OFSD OBSD LOFLom</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OAIS OBDI OLCK OOCl 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 bit in the PM overhead field (byte 3, bit 5) is "1" for at least 5 consecutive frames.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:CURRent?**

OLCK, OLCK (ODU - Backward Defect indication) 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.

OFSF, ODU-FSF (ODU - Forward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000001".

OBSF, ODU-BSF (ODU - Backward Signal Fail) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000001".

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 math the expected SAPI and/or DAPI. This alarm is only available when TIM SAPI or DAPI is enabled.

OFSF, ODU-FSD (ODU - Forward Signal Degrade) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000010"

OBSF, ODU-BSD (ODU - Backward Signal Degrade) is declared when the received FTFL byte 128 is "00000010".

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.

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:CURRent?

Response Syntax <Current>

Response(s) Current:
The response data syntax for <Current> is defined as a <NR1 NUMERIC 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.

Example(s)

- * SOUR:DATA:TEL:OTN:ALAR:ODU1:TYPE OAIS
- * SOUR:DATA:TEL:OTN:ALAR:ODU1 ON
- * FETC:DATA:TEL:OTN:ALAR:ODU1:CURR? OAIS

Returns the current alarm status.

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.

See Also

- * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE
- * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:HISTory?**

Description	<p>This query returns the history status of Optical Data Unit (ODU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:HISTory? <wsp>OBIP8 OBEI</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the type of ODU (Optical Data Unit) error. OBIP8, selects 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, selects 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>
Response Syntax	<code><History></code>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Optical Data Unit (ODU) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:ODU1:AMO 15* SOUR:DATA:TEL:OTN:ERR:ODU1:INJ* FETC:DATA:TEL:OTN:ERR:ODU1:HIST? OBIP8 <p>Returns the error history status.</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:ODU[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:ODU[1..n]:AMOUnt* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:ODU[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:SECOnds?**

Description This query returns the number of seconds within which Optical Data Unit (ODU) error occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:SECOnds? <wsp>OBIP8|OBEI

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8|OBEI.
Selects the type of ODU (Optical Data Unit) error. OBIP8, selects 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).
OBEI, selects 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.

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Data Unit (ODU) error.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:ODU1:INJ * FETC:DATA:TEL:OTN:ERR:ODU1:SEC? OBIP8 Returns the number of errored seconds.
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOUnt * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:CURRent?**

Description	<p>This query returns the current status of Optical Data Unit (ODU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:CURRent? <wsp>OBIP8 OBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the type of ODU (Optical Data Unit) error. OBIP8, selects 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, selects 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>
Response Syntax	<pre><Current></pre>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:CURREnt?**

Response(s) Current:
The response data syntax for <Current> 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.

Example(s) * SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:AMO 15
* SOUR:DATA:TEL:OTN:ERR:ODU1:INJ
* FETC:DATA:TEL:OTN:ERR:ODU1:CURR? OBIP8
Returns the current error status.

Note OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:AMOUnt
* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:COUNT?**

Description	<p>This query returns the count of Optical Data Unit (ODU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:COUNT?<wsp>OBIP8 OBEI</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the type of ODU (Optical Data Unit) error. OBIP8, selects 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, selects 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>
Response Syntax	<code><Count></code>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:COUNT?

Response(s)	Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of Optical Data Unit (ODU) error.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:ODU1:INJ * FETC:DATA:TEL:OTN:ERR:ODU1:COUN? OBIP8 Returns the error count.
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AMOUnt * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:RATE?**

Description	<p>This query returns the current rate of Optical Data Unit (ODU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:RATE? <wsp>OBIP8 OBEI</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the type of ODU (Optical Data Unit) error. OBIP8, selects 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, selects 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>
Response Syntax	<code><Rate></code>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of Optical Data Unit (ODU) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:ODU1:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:ODU1:AMO 15* SOUR:DATA:TEL:OTN:ERR:ODU1:INJ* FETC:DATA:TEL:OTN:ERR:ODU1:RATE? OBIP8 <p>Returns the error rate.</p>
Note	<p>OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOUnt* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:HISTory?****Description**

This query returns the history status of Optical Data Unit (ODU) alarm for the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:HISTory? <wsp> TLTC | TBDI
| TTIM | TBIAE | TIAE

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:HISTory?**

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

TLTC|TBDI|TTIM|TBIAE|TIAE.

Selects the history status of Optical Data Unit (ODU) alarm for the TCM level.

TLTC, selects TCMi - Loss of Tandem Connection as the history status of Optical Data Unit (ODU) alarm.

TBDI, selects TCMi - Backward Defect Indication as the history status of Optical Data Unit (ODU) alarm.

TTIM, selects TCMi - Trace Identifier Mismatch as the history status of Optical Data Unit (ODU) alarm.

TBIAE, selects TCMi - Backward Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.

TIAE, selects TCMi - Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:HISTory?****Response Syntax** <History>**Response(s)**

History:

The response data syntax for <History> 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.

Example(s)

* FETC:DATA:TEL:OTN:ALAR:ODU1:TCM1:HIST?
TLTC Returns the ALAR history.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:SEConds?**

Description

This query returns the number of seconds within which Optical Data Unit (ODU) alarm occurred for the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:SEConds? <wsp> TLTC |
TBDI | TTIM | TBIAE | TIAE

:FETCh[1..n]:DATA:TELeCom:OTN:ALARm: ODU[1..n]:TCM[1..n]:SEConds?

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

TLTC|TBDI|TTIM|TBIAE|TIAE.

Selects the number of seconds within which Optical Data Unit (ODU) alarm occurred for the TCM level.

TLTC, selects TCMi - Loss of Tandem Connection as the number of seconds within which Optical Data Unit (ODU) alarm occurred.

TBDI, selects TCMi - Backward Defect Indication as the number of seconds within which Optical Data Unit (ODU) alarm occurred.

TTIM, selects TCMi - Trace Identifier Mismatch as the number of seconds within which Optical Data Unit (ODU) alarm occurred.

TBIAE, selects TCMi - Backward Incoming Alignment Error as the number of seconds within which Optical Data Unit (ODU) alarm occurred.

TIAE, selects TCMi - Incoming Alignment Error as the number of seconds within which Optical Data Unit (ODU) alarm occurred.

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:TCM[1..n]:SEConds?

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Optical Data Unit (ODU) alarm.

Example(s) * FETC:DATA:TEL:OTN:ALAR:ODU1:TCM1:
CURR? TLTC Returns the number of seconds of Optical Data Unit (ODU) alarm.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:CURRent?****Description**

This query returns the current status of Optical Data Unit (ODU) alarm for the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:CURRent? <wsp> TLTC |
TBDI | TTIM | TBIAE | TIAE

:FETCh[1..n]:DATA:TELeom:OTN:ALARm: ODU[1..n]:TCM[1..n]:CURRent?

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

TLTC|TBDI|TTIM|TBIAE|TIAE.

Selects the current status of Optical Data Unit (ODU) alarm for the TCM level.

TLTC, selects TCMi - Loss of Tandem Connection as the history status of Optical Data Unit (ODU) alarm.

TBDI, selects TCMi - Backward Defect Indication as the history status of Optical Data Unit (ODU) alarm.

TTIM, selects TCMi - Trace Identifier Mismatch as the history status of Optical Data Unit (ODU) alarm.

TBIAE, selects TCMi - Backward Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.

TIAE, selects TCMi - Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.

**:FETCh[1..n]:DATA:TELeom:OTN:ALARm:
ODU[1..n]:TCM[1..n]:CURRENT?****Response Syntax** <Current>**Response(s)**

Current:

The response data syntax for <Current> 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.

Example(s)

* FETC:DATA:TEL:OTN:ALAR:ODU1:TCM1:SEC?
TLTC Returns the current status of Optical Data Unit (ODU) alarm.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:TCM[1..n]:HISTory?**

Description	<p>This query returns the history status of Optical Data Unit (ODU) error for the TCM level.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:TCM[1..n]:HISTory? <wsp>TBIP8 TBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI.</p> <p>Selects the history status of Optical Data Unit (ODU) error for the TCM level.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:TCM[1..n]:HISTory?****Response Syntax** <History>**Response(s)**

History:

The response data syntax for <History> 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.

Example(s)

* FETC:DATA:TEL:OTN:ERR:ODU1:TCM1:HIST?
TBIP8 Returns the ALAR history.

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:TCM[1..n]:SEConds?

Description

This query returns the number of seconds within which Optical Data Unit (ODU) error occurred for the TCM level.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:  
ODU[1..n]:TCM[1..n]:SEConds? <wsp>TBIP8 |  
TBEI
```

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8|TBEI.

Selects the number of seconds within which Optical Data Unit (ODU) error occurred for the TCM level.

TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.

TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.

**:FETCh[1..n]:DATA:TELecom:OTN:ERROr:
ODU[1..n]:TCM[1..n]:SEConds?****Response Syntax** <Seconds>**Response(s)** Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Optical Data Unit (ODU) error.**Example(s)** * FETC:DATA:TEL:OTN:ERR:ODU1:TCM1:CURR?
TBIP8 Returns the number of seconds of Optical Data Unit (ODU) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:TCM[1..n]:CURRent?**

Description	<p>This query returns the current status of Optical Data Unit (ODU) error for the TCM level.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:TCM[1..n]:CURRent? <wsp>TBIP8 TBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI.</p> <p>Selects the current status of Optical Data Unit (ODU) error for the TCM level.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:TCM[1..n]:CURRENT?****Response Syntax** <Current>**Response(s)**

Current:

The response data syntax for <Current> 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.

Example(s)

* FETC:DATA:TEL:OTN:ERR:ODU1:TCM1:SEC?
TBIP8 Returns the current status of Optical Data Unit (ODU) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:TCM[1..n]:COUNT?**

Description This query returns the count of Optical Data Unit (ODU) error for the TCM level.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:TCM[1..n]:COUNT?<wsp>TBIP8|
TBEI

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8|TBEI.

Selects the count of Optical Data Unit (ODU) error for the TCM level.

TBIP8, selects the TCMi - Bit Interleave Parity-8 as the count of Optical Data Unit (ODU) error.

TBEI, selects the TCMi - Backward Error Indication as the count of Optical Data Unit (ODU) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:TCM[1..n]:COUNT?****Response Syntax** <Count>**Response(s)** Count:
The response data syntax for <Current> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of Optical Data Unit (ODU) error.**Example(s)** * FETC:DATA:TEL:OTN:ERR:ODU1:TCM1:COUN?
TBIP8 Returns the count of Optical Data Unit (ODU) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:TCM[1..n]:RATE?**

Description

This query returns the current rate of Optical Data Unit (ODU) error for ODU1e/2e of the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:TCM[1..n]:RATE? <wsp>TBIP8|TBEI

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8|TBEI.

Selects the current rate of Optical Data Unit (ODU) error for ODU1e/2e of the TCM level.

TBIP8, selects the TCMi - Bit Interleave Parity-8 as the current rate of Optical Data Unit (ODU) error.

TBEI, selects the TCMi - Backward Error Indication as the current rate of Optical Data Unit (ODU) error.

**:FETCh[1..n]:DATA:TELecom:OTN:ERROr:
ODU[1..n]:TCM[1..n]:RATE?****Response Syntax** <Rate>**Response(s)**

Rate:

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the current rate of Optical Data Unit (ODU) error.

Example(s)

* FETC:DATA:TEL:OTN:ERR:ODU1:TCM1:RATE?
TBIP8 Returns the current rate of Optical Data Unit (ODU) error.

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:HISTory?

Description

This query returns the history status of Optical Data Unit (ODU) alarm for ODU1e/2e of the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TCM[1..n]:HISTory?<wsp>TLTC |
TBDI | TTIM | TBIAE | TIAE

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TCM[1..n]:HISTory?****Parameter(s)**

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

TLTC|TBDI|TTIM|TBIAE|TIAE.

Selects the history status of Optical Data Unit (ODU) alarm for the TCM level.

TLTC, selects TCMi - Loss of Tandem Connection as the history status of Optical Data Unit (ODU) alarm.

TBDI, selects TCMi - Backward Defect Indication as the history status of Optical Data Unit (ODU) alarm.

TTIM, selects TCMi - Trace Identifier Mismatch as the history status of Optical Data Unit (ODU) alarm.

TBIAE, selects TCMi - Backward Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.

TIAE, selects TCMi - Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TCM[1..n]:HISTory?**

Response Syntax <History>

Response(s)

History:

The response data syntax for <History> 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.

Example(s)

* FETC:DATA:TEL:OTN:ALAR:ODU1:E:TCM1:
HIST? TLTC Returns the ALAR history.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TCM[1..n]:SEConds?****Description**

This query returns the number of seconds within which Optical Data Unit (ODU) alarm occurred for ODU1e/2e of the TCM level.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:  
ODU[1..n]:E:TCM[1..n]:SEConds? <wsp> TLTC |  
TBDI | TTIM | TBIAE | TIAE
```

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TCM[1..n]:SEConds?**

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

TLTC|TBDI|TTIM|TBIAE|TIAE.

Selects the number of seconds within which Optical Data Unit (ODU) alarm occurred for the TCM level.

TLTC, selects TCMi - Loss of Tandem Connection as the number of seconds within which Optical Data Unit (ODU) alarm occurred.

TBDI, selects TCMi - Backward Defect Indication as the number of seconds within which Optical Data Unit (ODU) alarm occurred.

TTIM, selects TCMi - Trace Identifier Mismatch as the number of seconds within which Optical Data Unit (ODU) alarm occurred.

TBIAE, selects TCMi - Backward Incoming Alignment Error as the number of seconds within which Optical Data Unit (ODU) alarm occurred.

TIAE, selects TCMi - Incoming Alignment Error as the number of seconds within which Optical Data Unit (ODU) alarm occurred.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TCM[1..n]:SEConds?****Response Syntax** <Seconds>**Response(s)** Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Optical Data Unit (ODU) alarm.**Example(s)** * FETC:DATA:TEL:OTN:ALAR:ODU1:E:TCM1:
CURR? TLTC Returns the number of seconds of
Optical Data Unit (ODU) alarm.

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:CURRent?

Description

This query returns the number of seconds within which Optical Data Unit (ODU) alarm occurred for ODU1e/2e of the TCM level.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:  
ODU[1..n]:E:TCM[1..n]:CURRent?<wsp>TLTC |  
TBDI | TTIM | TBIAE | TIAE
```

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TCM[1..n]:CURRent?****Parameter(s)**

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

TLTC|TBDI|TTIM|TBIAE|TIAE.

Selects the current status of Optical Data Unit (ODU) alarm for the TCM level.

TLTC, selects TCMi - Loss of Tandem Connection as the history status of Optical Data Unit (ODU) alarm.

TBDI, selects TCMi - Backward Defect Indication as the history status of Optical Data Unit (ODU) alarm.

TTIM, selects TCMi - Trace Identifier Mismatch as the history status of Optical Data Unit (ODU) alarm.

TBIAE, selects TCMi - Backward Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.

TIAE, selects TCMi - Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.

:FETCh[1..n]:DATA:TELeom:OTN:ALARm: ODU[1..n]:E:TCM[1..n]:CURRENT?

Response Syntax <Current>

Response(s)

Current:

The response data syntax for <Current> 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.

Example(s)

* FETC:DATA:TEL:OTN:ALAR:ODU1:E:TCM1:SEC? TLTC Returns the current status of Optical Data Unit (ODU) alarm.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:TCM[1..n]:HISTory?****Description**

This query returns the history status of Optical Data Unit (ODU) error for ODU1e/2e of the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:TCM[1..n]:HISTory?<wsp>TBIP8|
TBEI

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8|TBEI.

Selects the history status of Optical Data Unit (ODU) error for the TCM level.

TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.

TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:TCM[1..n]:HISTory?**

Response Syntax <History>

Response(s) History:
The response data syntax for <History> 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.

Example(s) * FETC:DATA:TEL:OTN:ERR:ODU1:E:TCM1:HIST?
TBIP8 Returns the ALAR history.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERror:
ODU[1..n]:E:TCM[1..n]:SEConds?****Description**

This query returns the number of seconds within which Optical Data Unit (ODU) error occurred for the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERror:
ODU[1..n]:E:TCM[1..n]:SEConds?<wsp>TBIP8|
TBEI

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8|TBEI.

Selects the number of seconds within which Optical Data Unit (ODU) error occurred for the TCM level.

TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.

TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:TCM[1..n]:SEConds?**

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Optical Data Unit (ODU) error.

Example(s) * FETC:DATA:TEL:OTN:ERR:ODU1:E:TCM1:SEC?
TBIP8 Returns the number of seconds of Optical Data Unit (ODU) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:TCM[1..n]:CURRent?****Description**

This query returns the current status of Optical Data Unit (ODU) error for ODU1e/2e of the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:TCM[1..n]:CURRent?<wsp>TBIP8|
TBEI

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8|TBEI.

Selects the current status of Optical Data Unit (ODU) error for the TCM level.

TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.

TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:CURRent?

Response Syntax <Current>

Response(s)

Current:

The response data syntax for <Current> 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.

Example(s)

* FETC:DATA:TEL:OTN:ERR:ODU1:E:TCM1:
CURR? TBIP8 Returns the current status of
Optical Data Unit (ODU) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:TCM[1..n]:COUNT?**

Description	<p>This query returns the count of Optical Data Unit (ODU) error for ODU1e/2e of the TCM level.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:E:TCM[1..n]:COUNT? <wsp>TBIP8 TBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI.</p> <p>Selects the count of Optical Data Unit (ODU) error for the TCM level.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the count of Optical Data Unit (ODU) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the count of Optical Data Unit (ODU) error.</p>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:COUNT?

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Current> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of Optical Data Unit (ODU) error.

Example(s) * FETC:DATA:TEL:OTN:ERR:ODU1:E:TCM1:
COUN? TBIP8 Returns the count of Optical Data Unit (ODU) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERror:
ODU[1..n]:E:TCM[1..n]:RATE?**

Description	<p>This query returns the current rate of Optical Data Unit (ODU) error for ODU1e/2e of the TCM level.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERror: ODU[1..n]:E:TCM[1..n]:RATE?<wsp>TBIP8 TBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI.</p> <p>Selects the current rate of Optical Data Unit (ODU) error for ODU1e/2e of the TCM level.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the current rate of Optical Data Unit (ODU) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the current rate of Optical Data Unit (ODU) error.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:TCM[1..n]:RATE?**

Response Syntax <Rate>

Response(s) Rate:
The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.
Returns the current rate of Optical Data Unit (ODU) error.

Example(s) * FETC:DATA:TEL:OTN:ERR:ODU1:E:TCM1:RATE?
TBIP8 Returns the current rate of Optical Data Unit (ODU) error.

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:TCM[1..n]:TYPE****Description**

This command selects the type of Optical Data Unit (ODU) error for automated injection of the TCM level.

At *RST, this value is set to TBIP8.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:TCM[1..n]:TYPE<wsp>
TBIP8|TBEI

**:SOURce[1..n]:DATA:TELecom:OTN:ERRor:
ODU[1..n]:AUTomated:TCM[1..n]:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI.</p> <p>Selects the type of Optical Data Unit (ODU) error for automated injection.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of Optical Data Unit (ODU) error for automated injection.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of Optical Data Unit (ODU) error for automated injection.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1 TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1: TYPE? Returns TBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELecom:OTN:ERRor: ODU1:AUTomated:TCM[1..n]TYPE?</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:TCM[1..n]:TYPE?**

Description	This query returns the type of Optical Data Unit (ODU) error for automated injection of the TCM level. At *RST, this value is set to TBIP8.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:AUTomated:TCM[1..n]:TYPE?
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TCM[1..n]:TYPE?

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Optical Data Unit (ODU) error for the automated injection.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of Optical Data Unit (ODU) error for automated injection.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of Optical Data Unit (ODU) error for automated injection.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1 TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1: TYPE? Returns TBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU1:AUTomated:TCM[1..n]TYPE</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TCM[1..n]:TYPE****Description**

This command selects the type of Optical Data Unit (ODU) error for ODU1e/2 automated injection.

At *RST, this value is set to TBIP8.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TCM[1..n]:TYPE<wsp>
TBIP8|TBEI

**:SOURce[1..n]:DATA:TELecom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TCM[1..n]:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI.</p> <p>Selects the type of Optical Data Unit (ODU) error for automated injection.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of Optical Data Unit (ODU) error for automated injection.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of Optical Data Unit (ODU) error for automated injection.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1 TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1: TYPE? Returns TBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELecom:OTN:ERRor: ODU1:E:AUTomated:TCM[1..n]TYPE?</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TCM[1..n]:TYPE?**

Description	This query returns the type of Optical Data Unit (ODU) error for automated injection of the TCM level. At *RST, this value is set to TBIP8.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:AUTomated:TCM[1..n]:TYPE?
Response Syntax	<Error>

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:AUTomated:TCM[1..n]:TYPE?

Response(s)

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Optical Data Unit (ODU) error for the automated injection.

TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of Optical Data Unit (ODU) error for automated injection.

TBEI, selects the TCMi - Backward Error Indication as the type of Optical Data Unit (ODU) error for automated injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1
TYPE TBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
TYPE? Returns TBIP8

See Also

* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU1:E:AUTomated:TCM[1..n]TYPE?

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:TCM[1..n]:RATE****Description**

This command selects the rate of Optical Data Unit (ODU) error for automated injection of the TCM level.

At *RST, this value is set to BERRor.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:TCM[1..n]:RATE<wsp>
MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:TCM[1..n]:RATE**

Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Sets the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1: TYPE TBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1: RATE 1.0E-09 * SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1: RATE? Returns 1.0E-09</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU1:AUTomated:TCM[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU1:AUTomated:TCM[1..n]:RATE?</pre>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:TCM[1..n]:RATE?**

Description	<p>This query returns the rate of Optical Data Unit (ODU) error for automated injection of the TCM level.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:AUTomated:TCM[1..n]:RATE? <wsp> MAXimum MINimum</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current rate will be returned.</p>
Response Syntax	<Rate>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:TCM[1..n]:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Optical Data Unit (ODU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:RATE? Returns 1.0E-09</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:AUTomated:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:AUTomated:TCM[1..n]:RATE</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TCM[1..n]:RATE****Description**

This command selects the rate of Optical Data Unit (ODU) error for ODU1e/2 automated injection.

At *RST, this value is set to BERRor.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TCM[1..n]:RATE <wsp>
MAXimum | MINimum

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TCM[1..n]:RATE**

Parameter(s)

Rate:

The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum.

Sets the injection rate for the selected Optical Data Unit (ODU) error.

MAXimum is used to retrieve the instrument's greatest supported value.

MINimum is used to retrieve the instrument's smallest supported value.

Example(s)

```
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
TYPE TBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
RATE 1.0E-09
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
RATE? Returns 1.0E-09
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:AUTomated:TCM[1..n]:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:AUTomated:TCM[1..n]:RATE?
```


**:SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:AUTomated:TCM[1..n]:RATE?**

Description	<p>This query returns the rate of Optical Data Unit (ODU) error for ODU1e/2 automated injection.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:E:AUTomated:TCM[1..n]:RATE? <wsp> MAXimum MINimum</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current injected rate will be returned.</p>
Response Syntax	<p><Rate></p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TCM[1..n]:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <CHARACTER RESPONSE DATA> element. Returns the injection rate for the selected Optical Data Unit (ODU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:RATE? Returns 1.0E-09</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:E:AUTomated:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:E:AUTomated:TCM[1..n]:RATE?</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:TCM[1..n]:AUTomated****Description**

This command enables or disables the selected automated Optical Data Unit (ODU) error at the rate specified or continuously for the TCM level.

At *RST, this value is set to OFF.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:TCM[1..n]:AUTomated <wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:AUTomated

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Enables or disables the automated Optical Data Unit (ODU) error injection.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1: TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1: RATE 1.0E-09 * SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AUT ON * SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AUT? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU1:AUTomated:TCM[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU1:AUTomated:TCM[1..n]:RATE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU1:TCM[1..n]:AUTomated?</pre>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:TCM[1..n]:AUTomated?**

Description This query returns the status of automated Optical Data Unit (ODU) error injection for ODU1e/2e of the TCM level.

At *RST, this value is set to OFF.

Syntax :SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:TCM[1..n]:AUTomated?

Parameter(s) None

Response Syntax <Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:TCM[1..n]:AUTomated?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of automated Optical Data Unit (ODU) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:
TYPE OBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:
RATE 1.0E-09

* SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AUT
ON

* SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AUT?
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TCM[1..n]:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TCM[1..n]:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:TCM[1..n]:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:TCM[1..n]:AUTomated****Description**

This command enables or disables the selected automated Optical Data Unit (ODU) error at the rate specified or continuously for ODU1e/2e of the TCM level.

At *RST, this value is set to OFF.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:TCM[1..n]:AUTomated <wsp>
<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:TCM[1..n]:AUTomated**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Enables or disables the automated Optical Data Unit (ODU) error injection.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:TCM1:AUT ON</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AUT? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:E:AUTomated:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:E:AUTomated:TCM[1..n]:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:E:TCM[1..n]:AUTomated?</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:TCM[1..n]:AUTomated?**

Description	This query returns the status of automated Optical Data Unit (ODU) error injection for ODU1e/2e of the TCM level. At *RST, this value is set to OFF.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:TCM[1..n]:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:AUTomated?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of automated Optical Data Unit (ODU) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
TYPE OBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
RATE 1.0E-09

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:TCM1:AUT
ON

* SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AUT?
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:AUTomated:TCM[1..n]:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:AUTomated:TCM[1..n]:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:TCM[1..n]:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous**

Description This command enables or disables the continuously rate of automated Optical Data Unit (ODU) error injection of the TCM level.

At *RST, this value is set to OFF.

Syntax :SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous
<wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous

Parameter(s)

Set:

The program data syntax for the parameter is defined as a <Boolean Program Data> element.

Enables or disables the continuous rate of automated Optical Data Unit (ODU) error injection.

Example(s)

```
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:  
TYPE TBIP8  
SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:  
CONT ON  
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:  
CONT? Returns 1  
* SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1: AUT  
ON
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:  
ODU1:AUTomated:TCM[1..n]:TYPE  
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:  
ODU1:TCM[1..n]:AUTomated  
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:  
ODU1:AUTomated:TCM[1..n]:CONTInuous?
```

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:AUTomated:TCM[1..n]:
CONTInuous?**

Description	<p>This query returns the status of continuously rate of automated Optical Data Unit (ODU) error injection of the TCM level.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:AUTomated:TCM[1..n]:
CONTInuous?**

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuously rate of automated Optical Data Unit (ODU) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:
TYPE TBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:
CONT ON
* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:
CONT? Returns 1
* SOUR:DATA:TEL:OTN:ERR:ODU1:TCM1:AUT
ON

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TCM[1..n]:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:TCM[1..n]:AUTomated
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:AUTomated:TCM[1..n]:CONTInuous

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TCM[1..n]:
CONTInuous**

Description	<p>This command enables or disables the continuously rate of automated Optical Data Unit (ODU) error injection for non standard rates ODU1e/2e of the TCM level.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:AUTomated:TCM[1..n]:CONTInuous <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Enables or disables the continuous rate of automated Optical Data Unit (ODU) error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TCM[1..n]:
CONTInuous**

Example(s)

- * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
TYPE TBIP8
- * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
CONT ON
- * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
CONT? Returns 1
- * SOUR:DATA:TEL:OTN:ERR:ODU1:E:TCM1:AUT
ON

See Also

- * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:AUTomated:TCM[1..n]:TYPE
- * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:TCM[1..n]:AUTomated
- * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:AUTomated:TCM[1..n]:CONTInuous?

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TCM[1..n]:
CONTInuous?**

Description	This query returns the status of continuously rate of automated Optical Data Unit (ODU) error injection of the TCM level. At *RST, this value is set to OFF.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AUTomated:TCM[1..n]:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TCM[1..n]:
CONTInuous?**

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of continuously rate of automated Optical Data Unit (ODU) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
TYPE TBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
CONT ON

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:
CONT? Returns 1

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:TCM1:AUT
ON

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:AUTomated:TCM[1..n]:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:TCM[1..n]:AUTomated

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:E:AUTomated:TCM[1..n]:CONTInuous

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TCM[1..n]:RATE****Description**

This command selects the rate of Optical Data Unit (ODU) error for ODU1e/2 automated injection.

At *RST, this value is set to BERRor.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TCM[1..n]:RATE<wsp>
MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TCM[1..n]:RATE**

Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Sets the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:TYPE TBIP8* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:RATE 1.0E-09* SOUR:DATA:TEL:OTN:ERR:ODU1:AUT:TCM1:RATE? Returns 1.0E-09
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:AUTomated:TCM[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:AUTomated:TCM[1..n]:RATE?

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TCM[1..n]:RATE?**

Description	<p>This query returns the type of Optical Data Unit (ODU) error for automated injection of the TCM level.</p> <p>At *RST, this value is set to BERRor.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TCM[1..n]:RATE? <wsp> MAXimum MINimum</pre>
Parameter	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current injected rate will be returned.</p>
Response Syntax	<pre><Error></pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TCM[1..n]:RATE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Optical Data Unit (ODU) error for the automated injection.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of Optical Data Unit (ODU) error for automated injection.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of Optical Data Unit (ODU) error for automated injection.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TCM1:TYPE? Returns TBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:E:AUTomated:TCM[1..n]TYPE?</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:HISTory?**

Description	<p>This query returns the history status of Optical Payload Unit (OPU) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:HISTory? <wsp>OPLM OMSim</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPLM OMSim. Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OPLM, selects the OPU-PLM (Optical Payload Unit Payload Mismatch) as alarm type.</p> <p>OMSim, selects the OPU-MSIM (Optical Payload Unit Multiplex Structure Identifier Mismatch) as alarm type.</p>
Response Syntax	<History>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1 ON* FETC:DATA:TEL:OTN:ALAR:OPU1:HIST? OMS <p>Returns the alarm history status.</p>
Note	<ul style="list-style-type: none">* OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.* OMSim is available with ODU mux only.* For 8120NGE/8130NGE/8130NGEv2 modules, choices are OPLM OMSim.For 8140 module, choice is OPLM only.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:SEConds?**

Description	<p>This query returns the number of seconds within which Optical Payload Unit (OPU) alarm occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:SEConds?<wsp>OPLM OMSim</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPLM OMSim. Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OPLM, selects the OPU-PLM (Optical Payload Unit Payload Mismatch) as alarm type.</p> <p>OMSim, selects the OPU-MSIM (Optical Payload Unit Multiplex Structure Identifier Mismatch) as alarm type.</p>
Response Syntax	<pre><Seconds></pre>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Payload Unit (OPU) alarm.
Example(s)	* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS * SOUR:DATA:TEL:OTN:ALAR:OPU1 ON * FETC:DATA:TEL:OTN:ALAR:OPU1:SEC? OMS Returns the number of seconds of OPU alarm.
Note	* OTU1/OTU2 are available for FTB/IQS-8120NGE/ 8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only. * OMSim is available with ODU mux only. * For 8120NGE/8130NGE/8130NGEv2 modules, choices are OPLM OMSim. For 8140 module, choice is OPLM only.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:CURRent?**

Description	<p>This query returns the current status of Optical Payload Unit (OPU) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:CURRent? <wsp>OPLM OMSim</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPLM OMSim.</p> <p>Selects the type of OPU (Optical Payload Unit) alarm.</p> <p>OPLM, selects the OPU-PLM (Optical Payload Unit Payload Mismatch) as alarm type.</p> <p>OMSim, selects the OPU-MSIM (Optical Payload Unit Multiplex Structure Identifier Mismatch) as alarm type.</p>
Response Syntax	<pre><Current></pre>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:CURRent?

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of 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 is not running.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1 ON* FETC:DATA:TEL:OTN:ALAR:OPU1:CURR? OMS <p>Returns the current alarm status.</p>
Note	<ul style="list-style-type: none">* OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.* OMSim is available with ODU mux only.* For 8120NGE/8130NGE/8130NGEv2 modules, choices are OPLM OMSim.For 8140 module, choice is OPLM only.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:HISTory?****Description**

This query returns the history status of Forward Error Correction (FEC) error.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:HISTory?<wsp>FCCW|FUCW|
FCSYmb|FCBit

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

FCCW|FUCW|FCSYmb|FCBit.

Selects the type of FEC (Forward Error Correction) error.

FCCW, selects the FCCORR-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.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:FEC:HISTOrY?**

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.

Response Syntax <History>

Response(s) History:
The response data syntax for <History> 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.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:HISTory?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AMO 15
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:FEC:HIST?
FCCW
Returns the error history status.

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AMOUnt
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:FEC:SECOnds?**

Description This query returns the number of seconds within which Forward Error Correction (FEC) error occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:FEC:SECOnds? <wsp>FCCW|FUCW|
FCSYmb|FCBit

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FCCW|FUCW|FCSYmb|FCBit.
Selects the type of FEC (Forward Error Correction) error.
FCCW, selects the FCCORR-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.

**:FETCh[1..n]:DATA:TELecom:OTN:ERROr:
OTU[1..n]:FEC:SEConds?**

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.

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Forward Error Correction (FEC) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:SEConds?**

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AMO 15
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:FEC:SEC?
FCCW
Returns the number of errored seconds.

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:CURRent?**

Description This query returns the current status of Forward Error Correction (FEC) error.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:CURRent? <wsp>FCCW|FUCW|
FCSYmb|FCBit

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FCCW|FUCW|FCSYmb|FCBit.
Selects the type of FEC (Forward Error Correction) error.
FCCW, selects the FCCORR-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.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:FEC:CURRent?**

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.

Response Syntax <Current>

Response(s) Current:
The response data syntax for <Current> 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 is not running.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:CURREnt?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN:TYPE FCCW* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AMO 15* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:INJ* FETC:DATA:TEL:OTN:ERR:OTU1:FEC:CURREnt? FCCW Returns the current error status.
Note	OTU1/OTU2 are available for FTB/IQS-8120NGE/8130NGE/8130NGEv2 Transport Blazer and OTU3 is available for FTB/IQS-8140 Transport Blazer only.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:FEC:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:FEC:AMOUnt* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:FEC:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:COUNT?**

Description	<p>This query returns the count of Forward Error Correction (FEC) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:COUNT? <wsp>FCCW FUCW FCSYmb FCBit</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit.</p> <p>Selects the type of FEC (Forward Error Correction) error.</p> <p>FCCW, selects the FCCORR-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>

**:FETCh[1..n]:DATA:TELecom:OTN:ERROr:
OTU[1..n]:FEC:COUNT?**

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.

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of Forward Error Correction (FEC) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:COUNT?**

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AMO 15
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:FEC:COUN?
FCCW Returns the error count.

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:RATE?**

Description	<p>This query returns the current rate of Forward Error Correction (FEC) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:FEC:RATE?<wsp>FCCW FUCW FCSYmb FCBit</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit.</p> <p>Selects the type of FEC (Forward Error Correction) error.</p> <p>FCCW, selects the FCCORR-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>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:FEC:RATE?**

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.

Response Syntax <Rate>

Response(s) Rate:
The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.
Returns the current rate of Forward Error Correction (FEC) error.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:RATE?****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:MAN:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:AMO 15
* SOUR:DATA:TEL:OTN:ERR:OTU1:FEC:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:FEC:RATE?
FCCW Returns the error rate.

Note

OTU1/OTU2 are available for FTB/IQS-8120NGE/
8130NGE/8130NGEv2 Transport Blazer and OTU3
is available for FTB/IQS-8140 Transport Blazer
only.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:FEC:INJect

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:OVERhead**

Description	<p>This command sets the Optical Transport Unit (OTU) overhead byte values for the transmitter.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>: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</pre>
Parameter(s)	<p>Overhead:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OA111 OA112 OA113 OA214 OA215 OA216 MFAS17 SM18 SM19 SM110 GCC0111 GCC0112 RES113 RES114.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:OVERhead**

Selects OTU overhead bytes.

OA111, selects OA1 as overhead byte.

OA112, selects OA1 as overhead byte.

OA113, selects OA1 as overhead byte.

OA214, selects OA2 as overhead byte.

OA215, selects OA2 as overhead byte.

OA216, selects OA2 as overhead byte.

MFAS17, selects MFAS as overhead byte.

SM18, selects SM as overhead byte.

SM19, selects SM as overhead byte.

SM110, selects SM as overhead byte.

GCC0111, selects GCC0 as overhead byte.

GCC0112, selects GCC0 as overhead byte.

RES113, selects RES as overhead byte.

RES114, selects RES as overhead byte.

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.

Value:

The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the Optical Transport Unit (OUT) overhead byte values.

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:OVERhead

Example(s) * SOUR:DATA:TEL:OTN:OH:OTU1:OVER OA111,
 #HF6
 * SOUR:DATA:TEL:OTN:OH:OTU1:OVER? OA111
 Returns #HF6

See Also * SOURce[1..n]:DATA:TELEcom:OTN:OH:
 OTU[1..n]:OVERhead?

**: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, this value is device dependent.

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

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:OVERhead?**

Parameter(s)	Overhead:
	<p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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, selects OA1 as overhead byte. OA112, selects OA1 as overhead byte. OA113, selects OA1 as overhead byte. OA214, selects OA2 as overhead byte. OA215, selects OA2 as overhead byte. OA216, selects OA2 as overhead byte. MFAS17, selects MFAS as overhead byte. SM18, selects SM as overhead byte. SM19, selects SM as overhead byte. SM110, selects SM as overhead byte. GCC0111, selects GCC0 as overhead byte. GCC0112, selects GCC0 as overhead byte. RES113, selects RES as overhead byte. RES114, selects RES as overhead byte.</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>

**:SOURce[1..n]:DATA:TELecom:OTN:OH:
OTU[1..n]:OVERhead?****Response Syntax** <Value>**Response(s)**

Value:

The response data syntax for <Statistics> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the OTU overhead byte values in hexadecimal format.

Example(s)

* SOUR:DATA:TEL:OTN:OH:OTU1:OVER OA111,
#HF6

* SOUR:DATA:TEL:OTN:OH:OTU1:OVER? OA111
Returns #HF6

See Also

* SOURce[1..n]:DATA:TELecom:OTN:OH:
OTU[1..n]:OVERhead

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

Description	<p>This command resets or overwrites the Optical Transport Unit (OTU) overhead byte values.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:DEFault</p>
Parameter(s)	<p>None</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OH:OTU1:DEF</p>

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]
:OVERhead?****Description**

This query returns the Optical Transport Unit (OTU) overhead byte values for the receiver.

At *RST, this value is device dependent.

Syntax

```
:SENSe[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
```

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n]
:OVERhead?**

Parameter(s)	Overhead:
	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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, selects OA1 as overhead byte. OA112, selects OA1 as overhead byte. OA113, selects OA1 as overhead byte. OA214, selects OA2 as overhead byte. OA215, selects OA2 as overhead byte. OA216, selects OA2 as overhead byte. MFAS17, selects MFAS as overhead byte. SM18, selects SM as overhead byte. SM19, selects SM as overhead byte. SM110, selects SM as overhead byte. GCC0111, selects GCC0 as overhead byte. GCC0112, selects GCC0 as overhead byte. RES113, selects RES as overhead byte. RES114, selects RES as overhead byte.</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>

**:SENSe[1..n]:DATA:TELeom:OTN:OH:OTU[1..n]
:OVERhead?****Response Syntax** <Value>**Response(s)**

Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the OTU overhead byte values in hexadecimal format.

Example(s)

* SENS:DATA:TEL:OTN:OH:OTU1:OVER? OA111
Returns the OTU overhead byte values.

See Also

* SOURce[1..n]:DATA:TELeom:OTN:OH:
OTU[1..n]:OVERhead

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead

Description

This command sets the Optical Data Unit (ODU) overhead byte values for the transmitter.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:  
ODU[1..n]:OVERhead<wsp>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  
<Value>
```

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead

Parameter(s)	Overhead:
	<p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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 ODU overhead bytes.</p> <p>RES21, selects RES as overhead byte.</p> <p>RES22, selects RES as overhead byte.</p> <p>RES23, selects RES as overhead byte.</p> <p>TCMACT24, selects TCMACT as overhead byte.</p> <p>TCM625, selects TCM6 as overhead byte.</p> <p>TCM626, selects TCM6 as overhead byte.</p> <p>TCM627, selects TCM6 as overhead byte.</p> <p>TCM528, selects TCM5 as overhead byte.</p> <p>TCM529, selects TCM5 as overhead byte.</p> <p>TCM5210, selects TCM5 as overhead byte.</p>

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead

TCM4211, selects TCM4 as overhead byte.
TCM4212, selects TCM4 as overhead byte.
TCM4213, selects TCM4 as overhead byte.
FTFL214, selects FTFL2 as overhead byte.
TCM331, selects TCM3 as overhead byte.
TCM332, selects TCM3 as overhead byte.
TCM333, selects TCM3 as overhead byte.
TCM234, selects TCM2 as overhead byte.
TCM235, selects TCM2 as overhead byte.
TCM236, selects TCM2 as overhead byte.
TCM137, selects TCM1 as overhead byte.
TCM138, selects TCM1 as overhead byte.
TCM139, selects TCM1 as overhead byte.
PM310, selects PM3 as overhead byte.
PM311, selects PM3 as overhead byte.
PM312, selects PM3 as overhead byte.
EXP313, selects EXP3 as overhead byte.
EXP314, selects EXP3 as overhead byte.
GCC141, selects GCC1 as overhead byte.
GCC142, selects GCC1 as overhead byte.
GCC243, selects GCC2 as overhead byte.
GCC244, selects GCC2 as overhead byte.
APSPCC45, selects APSPCC as overhead byte.
APSPCC46, selects APSPCC as overhead byte.
APSPCC47, selects APSPCC as overhead byte.

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead

APSPCC48, selects APSPCC as overhead byte.

RES49, selects RES as overhead byte.

RES410, selects RES as overhead byte.

RES411, selects RES as overhead byte.

RES412, selects RES as overhead byte.

RES413, selects RES as overhead byte.

RES414, selects RES as overhead byte.

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.

Value:

The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the Optical Data Unit (ODU) overhead byte values.

Example(s)

* SOUR:DATA:TEL:OTN:OH:ODU1:OVER RES21,
#HF6

* SOUR:DATA:TEL:OTN:OH:ODU1:OVER? RES21
Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:OVERhead?

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead?

Description

This query returns the Optical Data Unit (ODU) overhead byte values for the transmitter.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:  
ODU[1..n]:OVERhead?<wsp>RES21 |  
RES22 | RES23 | TCM24 | 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
```

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead?

Parameter(s)

Overhead:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

RES21 | RES22 | RES23 | TCM24 | 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.

Selects ODU overhead bytes.

RES21, selects RES as overhead byte.

RES22, selects RES as overhead byte.

RES23, selects RES as overhead byte.

TCMACT24, selects TCMACT as overhead byte.

TCM625, selects TCM6 as overhead byte.

TCM626, selects TCM6 as overhead byte.

TCM627, selects TCM6 as overhead byte.

TCM528, selects TCM5 as overhead byte.

TCM529, selects TCM5 as overhead byte.

TCM5210, selects TCM5 as overhead byte.

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead?

TCM4211, selects TCM4 as overhead byte.
TCM4212, selects TCM4 as overhead byte.
TCM4213, selects TCM4 as overhead byte.
FTFL214, selects FTFL2 as overhead byte.
TCM331, selects TCM3 as overhead byte.
TCM332, selects TCM3 as overhead byte.
TCM333, selects TCM3 as overhead byte.
TCM234, selects TCM2 as overhead byte.
TCM235, selects TCM2 as overhead byte.
TCM236, selects TCM2 as overhead byte.
TCM137, selects TCM1 as overhead byte.
TCM138, selects TCM1 as overhead byte.
TCM139, selects TCM1 as overhead byte.
PM310, selects PM3 as overhead byte.
PM311, selects PM3 as overhead byte.
PM312, selects PM3 as overhead byte.
EXP313, selects EXP3 as overhead byte.
EXP314, selects EXP3 as overhead byte.
GCC141, selects GCC1 as overhead byte.
GCC142, selects GCC1 as overhead byte.
GCC243, selects GCC2 as overhead byte.
GCC244, selects GCC2 as overhead byte.
APSPCC45, selects APSPCC as overhead byte.
APSPCC46, selects APSPCC as overhead byte.
APSPCC47, selects APSPCC as overhead byte.

**:SOURce[1..n]:DATA:TELecom:OTN:OH:
ODU[1..n]:OVERhead?**

APSPCC48, selects APSPCC as overhead byte.

RES49, selects RES as overhead byte.

RES410, selects RES as overhead byte.

RES411, selects RES as overhead byte.

RES412, selects RES as overhead byte.

RES413, selects RES as overhead byte.

RES414, selects RES as overhead byte.

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.

Response Syntax <Value>

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead?

Response(s)	Value: The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the ODU overhead byte values in hexadecimal format.
Example(s)	* SOUR:DATA:TEL:OTN:OH:ODU1:OVER RES21, #HF6 * SOUR:DATA:TEL:OTN:OH:ODU1:OVER? RES21 Returns #HF6
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead

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

Description	<p>This command resets or overwrites the Optical Data Unit (ODU) overhead byte values.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:DEFault
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:OH:ODU1:DEF

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:OVERhead?**

Description This query returns the Optical Data Unit (ODU) overhead byte values for the receiver.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]
:OVERhead? <wsp> RES21 | RES22 | RES23 |
TCM24 | 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

:SENSe[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead?

Parameter(s)

Overhead:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

RES21 | RES22 | RES23 | TCM24 | 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.

Selects ODU overhead bytes.

RES21, selects RES as overhead byte.

RES22, selects RES as overhead byte.

RES23, selects RES as overhead byte.

TCMACT24, selects TCMACT as overhead byte.

TCM625, selects TCM6 as overhead byte.

TCM626, selects TCM6 as overhead byte.

TCM627, selects TCM6 as overhead byte.

TCM528, selects TCM5 as overhead byte.

TCM529, selects TCM5 as overhead byte.

TCM5210, selects TCM5 as overhead byte.

:SENSE[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:OVERhead?

TCM4211, selects TCM4 as overhead byte.
TCM4212, selects TCM4 as overhead byte.
TCM4213, selects TCM4 as overhead byte.
FTFL214, selects FTFL2 as overhead byte.
TCM331, selects TCM3 as overhead byte.
TCM332, selects TCM3 as overhead byte.
TCM333, selects TCM3 as overhead byte.
TCM234, selects TCM2 as overhead byte.
TCM235, selects TCM2 as overhead byte.
TCM236, selects TCM2 as overhead byte.
TCM137, selects TCM1 as overhead byte.
TCM138, selects TCM1 as overhead byte.
TCM139, selects TCM1 as overhead byte.
PM310, selects PM3 as overhead byte.
PM311, selects PM3 as overhead byte.
PM312, selects PM3 as overhead byte.
EXP313, selects EXP3 as overhead byte.
EXP314, selects EXP3 as overhead byte.
GCC141, selects GCC1 as overhead byte.
GCC142, selects GCC1 as overhead byte.
GCC243, selects GCC2 as overhead byte.
GCC244, selects GCC2 as overhead byte.
APSPCC45, selects APSPCC as overhead byte.
APSPCC46, selects APSPCC as overhead byte.
APSPCC47, selects APSPCC as overhead byte.

:SENSe[1..n]:DATA:TELeom:OTN:OH: ODU[1..n]:OVERhead?

APSPCC48, selects APSPCC as overhead byte.

RES49, selects RES as overhead byte.

RES410, selects RES as overhead byte.

RES411, selects RES as overhead byte.

RES412, selects RES as overhead byte.

RES413, selects RES as overhead byte.

RES414, selects RES as overhead byte.

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.

Response Syntax <Value>

Response(s)

Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the ODU overhead byte values in hexadecimal format.

Example(s)

* SENS:DATA:TEL:OTN:OH:ODU1:OVER? RES21
Returns the ODU overhead byte values.

See Also

* SOURce[1..n]:DATA:TELeom:OTN:OH:
ODU[1..n]:OVERhead

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:OVERhead

Description

This command sets the Optical Payload Unit (OPU) overhead byte values for the transmitter.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:  
OPU[1..n]:OVERhead<wsp>RES115|RES116|  
JC116|RES215|RES216|JC216|RES315|RES316|  
JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|  
PSI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|  
PSI13|PSI14|PSI15|PSI16|PSI17|JC1|JC2|JC3|  
JC4|JC5|JC6|RES416,<Value>
```

:SOURCE[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:OVERhead

Parameter(s)	Overhead:
	<p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>RES115 RES116 JC116 RES215 RES216 JC216 RES315 RES316 JC316 PSI415 NJO416 PSI0 PSI1 PSI2 PSI3 PSI4 PSI5 PSI6 PSI7 PSI8 PSI9 PSI10 PSI11 PSI12 PSI13 PSI14 PSI15 PSI16 PSI17 JC1 JC2 JC3 JC4 JC5 JC6 RES416.</p> <p>Selects OPU overhead bytes.</p> <p>RES115, selects RES as overhead byte.</p> <p>RES116, selects RES as overhead byte.</p> <p>JC116, selects JC as overhead byte.</p> <p>RES215, selects RES as overhead byte.</p> <p>RES216, selects RES as overhead byte.</p> <p>JC216, selects JC as overhead byte.</p> <p>RES315, selects RES as overhead byte.</p> <p>RES316, selects RES as overhead byte.</p> <p>JC316, selects JC as overhead byte.</p> <p>PSI415, selects PSI as overhead byte.</p> <p>NJO416, selects NJO as overhead byte.</p> <p>PSI0, selects PSI0 as overhead byte.</p> <p>PSI2, selects PSI2 as overhead byte.</p> <p>PSI3, selects PSI3 as overhead byte.</p>

**:SOURce[1..n]:DATA:TELecom:OTN:OH:
OPU[1..n]:OVERhead**

PSI4, selects PSI4 as overhead byte.

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

JC1, selects JC1 as overhead byte.

JC2, selects JC2 as overhead byte.

JC3, selects JC3 as overhead byte.

JC4, selects JC4 as overhead byte.

JC5, selects JC5 as overhead byte.

JC6, selects JC6 as overhead byte.

RES416, selects RES41 as overhead byte.

Note: The combination of row and column number is used along with overhead byte. For Ex: RES115, here RES is overhead byte, 1 is row number, and 15 is column number.

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:OVERhead**

Value:

The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the Optical Payload Unit (OPU) overhead byte values.

Example(s)

* SOUR:DATA:TEL:OTN:OH:OPU1:OVER RES115,
#HF6

* SOUR:DATA:TEL:OTN:OH:OPU1:OVER? RES115
Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:OVERhead?

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:OVERhead?**

Description This query returns the Optical Payload Unit (OPU) overhead byte values for the transmitter.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:OVERhead?<wsp>RES115|RES116|
JC116|RES215|RES216|JC216|RES315|RES316|
JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|
PSI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|
PSI13|PSI14|PSI15|PSI16|PSI17|JC1|JC2|JC3|
JC4|JC5|JC6|RES416.

Parameter(s) Overhead:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
RES115|RES116|JC116|RES215|RES216|JC216|
RES315|RES316|JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|PSI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|PSI13|PSI14|PSI15|PSI16|PSI17|JC1|JC2|JC3|JC4|JC5|JC6|RES416..

**:SOURce[1..n]:DATA:TELecom:OTN:OH:
OPU[1..n]:OVERhead?**

Selects OPU overhead bytes.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:OVERhead?

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

JC1, selects JC1 as overhead byte.

JC2, selects JC2 as overhead byte.

JC3, selects JC3 as overhead byte.

JC4, selects JC4 as overhead byte.

JC5, selects JC5 as overhead byte.

JC6, selects JC6 as overhead byte.

RES416, selects RES41 as overhead byte.

Note: The combination of row and column number is used along with overhead byte. For Ex: RES115, here RES is overhead byte, 1 is row number, and 15 is column number.

Response Syntax <Value>

**:SOURce[1..n]:DATA:TELecom:OTN:OH:
OPU[1..n]:OVERhead?****Response(s)**

Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the Optical Payload Unit (OPU) overhead byte values for the transmitter.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:OVERhead?**

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Example(s)

* SOUR:DATA:TEL:OTN:OH:OPU1:OVER RES115,
#HF6

* SOUR:DATA:TEL:OTN:OH:OPU1:OVER? RES115
Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:OVERhead

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

Description	<p>This command resets or overwrites the Optical Payload Unit (OPU) overhead byte values.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:DEFault
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:OH:OPU1:DEF

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

Description This query returns the Optical Payload Unit (OPU) overhead byte values for the receiver.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:OVERhead? <wsp> RES115|RES116|JC116|RES215|RES216|JC216|RES315|RES316|JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|PSI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|PSI13|PSI14|PSI15|PSI16|PSI17|JC1|JC2|JC3|JC4|JC5|JC6|RES416

Parameter(s) Overhead:
The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
RES115|RES116|JC116|RES215|RES216|JC216|RES315|RES316|JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|PSI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|PSI13|PSI14|PSI15|PSI16|PSI17|JC1|JC2|JC3|JC4|JC5|JC6|RES416 .

**:SENSe[1..n]:DATA:TELecom:OTN:OH:
OPU[1..n]:OVERhead?**

Selects OPU overhead bytes.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

**:SENSe[1..n]:DATA:TELecom:OTN:OH:
OPU[1..n]:OVERhead?**

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Note: The combination of row and column number is used along with overhead byte. For Ex: RES115, here RES is overhead byte, 1 is row number, and 15 is column number.

Response Syntax <Value>

**:SENSe[1..n]:DATA:TELeCom:OTN:OH:
OPU[1..n]:OVERhead?****Response(s)**

Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the Optical Payload Unit (OPU) overhead byte values for the receiver.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

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

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

JC1, selects JC1 as overhead byte.

JC2, selects JC2 as overhead byte.

JC3, selects JC3 as overhead byte.

JC4, selects JC4 as overhead byte.

JC5, selects JC5 as overhead byte.

JC6, selects JC6 as overhead byte.

RES416, selects RES416 as overhead byte.

Example(s)

* SENS:DATA:TEL:OTN:OH:OPU1:OVER? RES115
Returns the OPU overhead byte values.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:OVERhead

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

Description	<p>This command enables or disables the Forward Error Correction (FEC) for non standard rates OTU1e/2e of the transmitter.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:FEC <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Forward Error Correction (FEC) for the transmitter.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:E:FEC ON * SOUR:DATA:TEL:OTN:OTU1:E:FEC? Returns 1</pre>
See Also	<pre>* SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:FEC?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:FEC?

Description	<p>This query returns the status of Forward Error Correction (FEC) for non standard rates OTU1e/2e of the transmitter.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:FEC?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of Forward Error Correction (FEC) for the transmitter.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OTU1:E:FEC ON</p> <p>* SOUR:DATA:TEL:OTN:OTU1:E:FEC? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:FEC</p>

**:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:
FEC**

Description	<p>This command enables or disables the Forward Error Correction (FEC) for non standard rates OTU1e/2e of the receiver.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: FEC<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Forward Error Correction (FEC) for the receiver.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:OTU1:E:FEC ON * SENS:DATA:TEL:OTN:OTU1:E:FEC? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: FEC?</pre>

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

Description	<p>This query returns the status of Forward Error Correction (FEC) for non standard rates OTU1e/2e of the receiver.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<code>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: FEC?</code>
Parameter(s)	None
Response Syntax	<code><Set></code>
Response(s)	<p>Set:</p> <p>The response data syntax for <code><Set></code> is defined as a <code><NR1 NUMERIC RESPONSE DATA></code> element.</p> <p>Returns the status of Forward Error Correction (FEC) for the receiver.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OTU1:E:FEC ON</p> <p>* SENS:DATA:TEL:OTN:OTU1:E:FEC? Returns 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: FEC</p>

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

Description	<p>This command enables or disables the scrambler for non standard rates OTU1e/2e of the transmitter.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SCRAMbler<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the scrambler of the transmitter.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:E:SCR ON * SOUR:DATA:TEL:OTN:OTU1:E:SCR? Returns 1</pre>
See Also	<pre>* SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SCRAMbler?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SCRamblEr?

Description	<p>This query returns the status of scrambler for non standard rates OTU1e/2e of the transmitter.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SCRamblEr?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the scrambler for the transmitter.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OTU1:E:SCR ON</p> <p>* SOUR:DATA:TEL:OTN:OTU1:E:SCR? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:SCRamblEr</p>

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

Description	<p>This command enables or disables the scrambler for non standard rates OTU1e/2e of the receiver.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SCRambler<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the scrambler for the receiver.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:OTU1:E:SCR ON * SENS:DATA:TEL:OTN:OTU1:E:SCR? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SCRambler?</pre>

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

Description	<p>This query returns the status of the scrambler for non standard rates OTU1e/2e of the receiver.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SCRambler?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the scrambler for the receiver.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OTU1:E:SCR ON</p> <p>* SENS:DATA:TEL:OTN:OTU1:E:SCR? Returns 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SCRambler</p>

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

Description	<p>This command sets the injected message for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SM:SAPI:B16<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:E:SM:SAPI:B16 "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:OTU1:E:SM:SAPI:B16? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SOURCE[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SM:SAPI:B16?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SM:SAPI:B16?

Description	This query returns the injected message for non standard rates OTU1e/2e. At *RST, this value is set to "EXFO OTU SAPI".
Syntax	: SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SM:SAPI:B16?
Parameter(s)	None
Response Syntax	<Message>

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:
SM:SAPI:B16?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the selected message.

Example(s)

* SOUR:DATA:TEL:OTN:OTU1:E:SM:SAPI:B16
"EXFO OTU SAPI"

* SOUR:DATA:TEL:OTN:OTU1:E:SM:SAPI:B16?
Returns "EXFO OTU SAPI"

See Also

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

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

Description This command sets the injected message for non standard rates OTU1e/2e.

At *RST, this value is set to "EXFO OTU DAPI".

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
E:SM:DAPI:B16<wsp><Message>

Parameter(s) Message:
The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.
Sets the selected message.

Example(s) * SOUR:DATA:TEL:OTN:OTU1:E:SM:DAPI:B16
"EXFO OTU DAPI"
* SOUR:DATA:TEL:OTN:OTU1:E:SM:DAPI:B16?
Returns "EXFO OTU DAPI"

See Also * SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]
:E:SM:DAPI:B16?

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

Description	This query returns the injected message for non standard rates OTU1e/2e. At *RST, this value is set to "EXFO OTU DAPI".
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n] :E:SM:DAPI:B16?
Parameter(s)	None
Response Syntax	<Message>

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SM:DAPI:B16?

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the selected message.
Example(s)	* SOUR:DATA:TEL:OTN:OTU1:E:SM:DAPI:B16 "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:OTU1:E:SM:DAPI:B16? Returns "EXFO OTU DAPI"
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n] :E:SM:DAPI:B16

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

Description	<p>This command sets the injected message for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E :SM:OPSPec:B32<wsp><Message></pre>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the selected message.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:E:SM:OPSP:B32 "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:OTU1:E:SM:OPSP:B32? Returns "EXFO OTU OPERATOR SPECIFIC"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SM:OPSPec:B32?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SM:OPSPec:B32?

Description	This query returns the injected message for non standard rates OTU1e/2e. At *RST, this value is set to "EXFO OTU OPERATOR SPECIFIC".
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SM:OPSPec:B32?
Parameter(s)	None
Response Syntax	<Message>

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:
SM:OPSPec:B32?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the selected message.

Example(s)

* SOUR:DATA:TEL:OTN:OTU1:E:SM:OPSP:B32
"EXFO OTU OPERATOR SPECIFIC"

* SOUR:DATA:TEL:OTN:OTU1:E:SM:OPSP:B32?
Returns "EXFO OTU OPERATOR SPECIFIC"

See Also

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

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SM:OVERwrite:ENABLEd

Description	<p>This command enables or disables the SM Overwrite feature for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: SM:OVERwrite:ENABLEd<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the SM Overwrite feature.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:E:SM:OVER:ENAB ON * SOUR:DATA:TEL:OTN:OTU1:E:SM:OVER:ENAB? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SM:OVERwrite:ENABLEd?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:
SM:OVERwrite:ENABLEd?**

Description	<p>This query returns the status of the SM Overwrite feature for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SM:OVERwrite:ENABLEd?</pre>
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the SM Overwrite feature.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:E:SM:OVER:ENAB ON * SOUR:DATA:TEL:OTN:OTU1:E:SM:OVER:ENAB? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: E:SM:OVERwrite:ENABLEd</pre>

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

Description	<p>This command sets the expected message for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:SAPI:EXPeCted <wsp> <Message></pre>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the expected message.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:OTU1:E:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:OTU1:E:TTI:SAPI:EXP "EXFO OTU SAPI" * SENS:DATA:TEL:OTN:OTU1:E:TTI:SAPI:EXP? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:TIM * SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:SAPI:EXPeCted?</pre>

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

Description	This query returns the expected message for non standard rates OTU1e/2e. At *RST, this value is set to "EXFO OTU SAPI".
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:SAPI:EXPeCted?
Parameter(s)	None
Response Syntax	<Message>

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

Response(s)

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the expected message.

Example(s)

* SENS:DATA:TEL:OTN:OTU1:E:TTI:TIM SAPI,ON

* SENS:DATA:TEL:OTN:OTU1:E:TTI:SAPI:EXP

"EXFO OTU SAPI"

* SENS:DATA:TEL:OTN:OTU1:E:TTI:SAPI:EXP?

Returns "EXFO OTU SAPI"

See Also

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

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

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

Description	<p>This command sets expected message for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:SAPI:EXPeCted <wsp> <Message></pre>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the expected message.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:OTU1:E:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:OTU1:E:TTI:DAPI:EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:OTU1:E:TTI:DAPI:EXP? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:TIM * SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:DAPI:EXPeCted?</pre>

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

Description	This query returns expected message for non standard rates OTU1e/2e. At *RST, this value is set to "EXFO OTU DAPI".
Syntax	SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:DAPI:EXPeCted?
Parameter(s)	None
Response Syntax	<Message>

**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:E:
TTI:DAPI:EXPeCted?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the expected message.

Example(s)

* SENS:DATA:TEL:OTN:OTU1:E:TTI:TIM DAPI,ON

* SENS:DATA:TEL:OTN:OTU1:E:TTI:DAPI:EXP

"EXFO OTU DAPI"

* SENS:DATA:TEL:OTN:OTU1:E:TTI:DAPI:EXP?

Returns "EXFO OTU DAPI"

See Also

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

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

:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:SAPI:B16?

Description	<p>This query returns the received message for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:SAPI:B16?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OTU1:E:TTI:SAPI:B16? Returns "EXFO OTU SAPI"</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E:
TTI:DAPI:B16?**

Description	<p>This query returns the received message for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:DAPI:B16?
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OTU1:E:TTI:DAPI:B16?</p> <p>Returns "EXFO OTU DAPI"</p>

:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:OPSPec:B32?

Description	<p>This query returns the received message for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:OPSPec:B32?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OTU1:E:TTI:OPSP:B32? Returns "EXFO OTU OPERATOR SPECIFIC"</p>

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

Description	<p>This command enables or disables the state of Trace Identifier Mismatch (TIM) for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:TIM<wsp>SAPI DAPI,<Set></pre>
Parameter(s)	<p>Etim:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SAPI DAPI.</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p> <p>SAPI, selects the SAPI which allows editing of the Source Access Point Identifier (SAPI) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the Destination Access Point Identifier (DAPI) message to be generated.</p>

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

Set:

The program data syntax for <Set> is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability.

ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the Trace Identifier Mismatch (TIM).

Example(s)

* SENS:DATA:TEL:OTN:OTU1:E:TTI:TIM SAPI,ON

* SENS:DATA:TEL:OTN:OTU1:E:TTI:TIM? SAPI

Returns 1

See Also

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

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

Description	<p>This query returns status of Trace Identifier Mismatch (TIM) for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to ON.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:E: TTI:TIM? <wsp> SAPI DAPI
Parameter(s)	<p>Etim:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SAPI DAPI.</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p> <p>SAPI, selects the SAPI which allows editing of the Source Access Point Identifier (SAPI) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the Destination Access Point Identifier (DAPI) message to be generated.</p>
Response Syntax	<Set>

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

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of the Trace Identifier Mismatch (TIM).

Example(s)

* SENS:DATA:TEL:OTN:OTU1:E:TTI:TIM SAPI,ON

* SENS:DATA:TEL:OTN:OTU1:E:TTI:TIM? SAPI

Returns 1

See Also

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

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:TTI:OVERwrite:ENABLEd**

Description	<p>This command enables or disables the Trail Trace Identifier (TTI) Overwrite feature for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:OVERwrite:ENABLEd <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Trail Trace Identifier (TTI) Overwrite feature.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:E:TTI:OVER:ENAB ON * SOUR:DATA:TEL:OTN:ODU1:E:TTI:OVER: ENAB? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:OVERwrite:ENABLEd?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:OVERwrite:ENABLEd?

Description	This query returns the status of the Trail Trace Identifier (TTI) Overwrite feature for non standard rates OTU1e/2e. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:OVERwrite:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:TTI:OVERwrite:ENABLEd?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of the Trail Trace Identifier (TTI) Overwrite feature.
Example(s)	* SOUR:DATA:TEL:OTN:ODU1:E:TTI:OVER:ENAB ON * SOUR:DATA:TEL:OTN:ODU1:E:TTI:OVER: ENAB? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:TTI:OVERwrite:ENABLEd

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

Description

This command sets the FTFL Fault Indication message to be generated for non standard rates OTU1e/2e.

At *RST, this value is set to NFAult.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:
:FTFL:INDication<wsp>FORWARD|BACKward,
NFAult|SFAil|SDEGrade|REServed

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

Parameter(s)**Ftfl:**

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWard|BACKward.

Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).

FORWard, sets the Forward configuration.

BACKward, sets the Backward configuration.

Indication:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
NFAult|SFAil|SDEGrade|REServed.

Sets the FTFL Fault Indication message to be generated.

NFAult, selects the No Fault (NFAult) as fault indication message.

SFAil, selects the Signal Fail (SFAil) as fault indication message.

SDEGrade, selects the Signal Degraded (SDEGrade) as fault indication message.

REServed, selects the Reserved as fault indication message.

:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: E:FTFL:INDication

Example(s)

* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:IND
FORW,SFA

* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:IND?
FORW Returns SFAIL

See Also

* SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]:
E:FTFL:INDication?

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

Description	<p>This query returns the FTFL Fault Indication message to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to NFAult.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:FTFL:INDication? <wsp>FORWARD BACKWARD</pre>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>
Response Syntax	<pre><Indication></pre>

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

Response(s)

Indication:

The response data syntax for <Indication> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the FTFL Fault Indication message to be generated.

NFAULT, No Fault (NFAULT) is selected as FTFL Fault Indication message.

SFAIL, Signal Fail (SFAIL) is selected as FTFL Fault Indication message.

SDEGRADE, Signal Degraded (SDEGRADE) is selected as FTFL Fault Indication message.

RESERVED, Reserved is selected as FTFL Fault Indication message.

Example(s)

* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:IND
FORW,SFA

* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:IND?
FORW Returns SFAIL

See Also

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

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

This command sets the FTFL Fault Indication code to be generated for non standard rates OTU1e/2e.

At *RST, this value is set to #H00.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E
:FTFL:CODE<wsp>FORWard|BACKward,
<Code>

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

Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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> <p>Code:</p> <p>The program data syntax for the second parameter is defined as a < NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the FTFL Fault Indication Code to be generated.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:CODE FORW,#H01</p> <p>* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:CODE? FORW Returns #H01</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:CODE?</p>

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

Description	<p>This query returns the FTFL Fault Indication code to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to #H00.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E :FTFL:CODE? <wsp>FORWard BACKward</pre>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>
Response Syntax	<pre><Code></pre>

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

Response(s)

Code:

The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the selection of the Fault Type Fault Location (FTFL) fault indication code to be generated.

Example(s)

* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:CODE FORW,#H01

* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:CODE? FORW Returns #H01

See Also

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

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

This command sets the FTFL Operator Identifier to be generated for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:
:FTFL:IDENtifier <wsp>FORWard|BACKward,
<Identifier>

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

Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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> <p>Identifier:</p> <p>The program data syntax for the second parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the operator identifier to be generated.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:IDEN FORW,"exfo"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:IDEN? FORW Returns "exfo"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:IDENtifier?</p>

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

Description	<p>This query returns the FTFL Operator Identifier to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:FTFL:IDENtifier? <wsp>FORWARD BACKWARD</pre>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>
Response Syntax	<pre><Identifier></pre>

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

Response(s)	<p>Identifier:</p> <p>The response data syntax for <Identifier> is defined as a <STRING RESPONSE DATA> 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>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:IDEN FORW,"exfo"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:IDEN? FORW Returns "exfo"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:IDENtifier</p>

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

Description	<p>This command sets the FTFL Operator Specific to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to "EXFO ODU OPERATOR SPECIFIC".</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E :FTFL:OPSPec<wsp>FORWARD BACKward, <Specific></pre>

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

Parameter(s)

Ftfl:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWard | BACKward.

Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).

FORWard, sets the Forward configuration.

BACKward, sets the Backward configuration.

Specific:

The program data syntax for the second parameter is defined as a <STRING PROGRAM DATA> element.

Sets the operator specific to be generated.

Example(s)

```
* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:OPSP  
FORW,"exfo"
```

```
* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:OPSP?  
FORW Returns "exfo"
```

See Also

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

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

Description	<p>This query returns the FTFL Operator Specific to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to "EXFO ODU OPERATOR SPECIFIC".</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E:FTFL:OPSPec?<wsp>FORWard BACKward
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>
Response Syntax	<Specific>

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

Response(s)	<p>Specific:</p> <p>The response data syntax for <Specific> is defined as a <STRING RESPONSE DATA> 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>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:OPSP FORW,"exfo"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:OPSP? FORW Returns "exfo"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:OPSPec</p>

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

Description	<p>This query returns the FTFL Fault Indication message to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: FTFL:INDication?<wsp>FORWARD BACKWARD</p>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>

:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: E:FTFL:INDIcation?

Response Syntax <Indication>

Response(s)

Indication:

The response data syntax for <Indication> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the FTFL Fault Indication message to be generated.

NFAULT, No Fault (NFAULT) is selected as FTFL Fault Indication message.

SFAIL, Signal Fail (SFAIL) is selected as FTFL Fault Indication message.

SDEGRADE, Signal Degraded (SDEGRADE) is selected as FTFL Fault Indication message.

RESERVED, Reserved is selected as FTFL Fault Indication message.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:E:FTFL:IND?
FORW Returns the FTFL Fault Indication message to be generated.

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

Description	<p>This query returns the FTFL Fault Indication code to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: FTFL:CODE?<wsp>FORWARD BACKWARD</p>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>

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

Response Syntax <Code>

Response(s) Code:
The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.
Returns the FTFL Fault Indication Code to be generated.

Example(s) * SENS:DATA:TEL:OTN:ODU1:E:FTFL:CODE?
FORW Returns the FTFL Fault Indication code to be generated.

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:FTFL:IDENTifier?**

Description	<p>This query returns the FTFL Operator Identifier to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: FTFL:IDENTifier?<wsp>FORWard BACKward</p>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:
E:FTFL:IDENTifier?**

Response Syntax <Identifier>

Response(s)

Identifier:

The response data syntax for <Identifier> is defined as a <STRING RESPONSE DATA> element.

Returns the value of the FTFL Operator Identifier (bytes 1 to 9 for forward, byte 129 to 137 for backward) to be generated.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:E:FTFL:IDEN?
FORW Returns the FTFL Operator Identifier to be generated.

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:FTFL:SPECific?**

Description	<p>This query returns the FTFL Operator Specific to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: FTFL:SPECific? <wsp>FORWARD BACKWARD</p>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>

:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]: E:FTFL:SPECific?

Response Syntax <Specific>

Response(s) Specific:
The response data syntax for <Specific> is defined as a <STRING RESPONSE DATA> element.
Returns the value of FTFL Operator Specific (bytes 10 to 127 for forward, byte 138 to 255 for backward) to be generated.

Example(s) * SENS:DATA:TEL:OTN:ODU1:E:FTFL:SPEC?
FORW Returns the FTFL Operator Specific to be generated.

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:FTFL:OVERwrite:ENABLEd**

Description	<p>This command enables or disables the Fault Type Fault Location (FTFL) Overwrite feature for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:E: :FTFL:OVERwrite:ENABLEd<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Fault Type Fault Location (FTFL) Overwrite feature.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:OVER: ENAB ON * SOUR:DATA:TEL:OTN:ODU1:E:FTFL:OVER: ENAB? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:OVERwrite:ENABLEd?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:OVERwrite:ENABLEd?

Description	This query returns the status of the Fault Type Fault Location (FTFL) Overwrite feature for non standard rates OTU1e/2e. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:OVERwrite:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
E:FTFL:OVERwrite:ENABLEd?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of the Fault Type Fault Location (FTFL) Overwrite feature.
Example(s)	* SOUR:DATA:TEL:OTN:ODU1:E:FTFL:OVER: ENAB ON * SOUR:DATA:TEL:OTN:ODU1:E:FTFL:OVER: ENAB? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: E:FTFL:OVERwrite:ENABLEd

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPE

Description	<p>This command sets the injected payload signal type to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to PRBStest.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E :PTYPE<wsp>EXPerimental ASYNchronous BISYNch ATM GFP1 VCONcate BSTiming BSNTiming ODUMux RFSTandard RPRopriet NULLtest PRBStest NAVailable</pre>
Parameter(s)	<p>Payload:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: EXPerimental ASYNchronous BISYNch ATM GFP1 VCONcate BSTiming BSNTiming ODUMux RFSTandard RPRopriet NULLtest PRBStest NAVailable.</p>

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

Sets the expected payload signal type to be generated.

EXPerimental, selects the payload type as Experimental.

ASYNchronous, selects the payload type as Asynchronous.

BISYNch, selects the payload type as Bit Synchronous (BISYNch).

ATM, selects the payload type as Asynchronous Transfer Mode (ATM).

GFP1, selects the payload type as Generic Framing Procedure (GFP).

VCONcate, selects the payload type as Virtual Concatenation (VCONcate).

BSTiming, selects the payload type as Bit Stream Timing (BSTiming).

BSNTiming, selects the payload type as Bit Stream No Timing (BSNTiming).

ODUMux, selects the payload type as Optical Data Unit Mux (ODUMux).

RFSTandard, selects the payload type as Reserved Future Standardization (RFSTandard).

RPRopriet, selects the payload type as Reserved Proprietary (RPRopriet).

NULLtest, selects the payload type as NULL Test.

PRBStest, selects the payload type as Pseudo Random Bit Sequence Test (PRBStest).

NAVailable, selects the payload type as Not Available.

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPE

Example(s)

* SOUR:DATA:TEL:OTN:OPU1:E:PTYP EXP

* SOUR:DATA:TEL:OTN:OPU1:E:PTYP?

Returns EXPERIMENTAL

Note

The Code field is automatically updated when the injected payload type is changed and vice versa.

See Also

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

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

Description	<p>This query returns the injected payload signal type to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to PRBStest.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPE?
Parameter(s)	None
Response Syntax	<Payload>
Response(s)	<p>Payload:</p> <p>The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the injected payload signal type to be generated</p> <p>EXPERIMENTAL, Experimental is selected as payload type.</p> <p>ASYNCHRONOUS, Asynchronous is selected as payload type.</p> <p>BISYNCH, Bit Synchronous (BISYNCH) is selected as payload type.</p>

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

ATM, Asynchronous Transfer Mode (ATM) is selected as payload type.

GFP1, Generic Framing Procedure (GFP) is selected as payload type.

VCONCATENATE, Virtual Concatenation (VCONCATENATE) is selected as payload type.

BSTIMING, Bit Stream Timing (BSTIMING) is selected as payload type.

BSNTIMING, Bit Stream No Timing (BSNTIMING) is selected as payload type.

ODUMUX, Optical Data Unit (ODU) Mux is selected as payload type.

RFSTANDARD, Reserved Future Standardization (RFSTANDARD) is selected as payload type.

RPROPRIET, Reserved Proprietary (RPROPRIET) is selected as payload type.

NULLTEST, NULL Test is selected as payload type.

PRBSTEST, Pseudo Random Bit Sequence Test (PRBSTEST) is selected as payload type.

NAVAILABLE, Not available is selected as payload type.

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

Example(s)	* SOUR:DATA:TEL:OTN:OPU1:E:PTYP EXP * SOUR:DATA:TEL:OTN:OPU1:E:PTYP? Returns EXPERIMENTAL
Note	The Code field is automatically updated when the injected payload type is changed and vice versa.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPe?

:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E: PTYPe:RECEived?

Description This query returns the received payload signal type to be generated for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:
:PTYPe:RECEived?

Parameter(s) None

Response Syntax <Payload>

**:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:
PTYPe:RECeived?****Response(s)**

Payload:

The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the received payload type.

EXPERIMENTAL, Experimental is retrieved.

ASYNCHRONOUS, Asynchronous is retrieved.

BISYNCH, Bit Synchronous (BISYNCH) is retrieved.

ATM, Asynchronous Transfer Mode (ATM) is retrieved.

GFP1, Generic Framing Procedure (GFP) is retrieved.

VCONCATENATE, Virtual Concatenation (VCONCATENATE) is retrieved.

BSTIMING, Bit Stream Timing (BSTIMING) is retrieved.

BSNTIMING, Bit Stream No Timing (BSNTIMING) is retrieved.

ODUMUX, Optical Data Unit (ODU) Mux is retrieved.

RFSTANDARD, Reserved Future Standardization (RFSTANDARD) is retrieved.

RPROPRIET, Reserved Proprietary (RPROPRIET) is retrieved.

NULLTEST, NULL Test is retrieved.

:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E: PTYPe:RECEived?

PRBSTEST, Pseudo Random Bit Sequence Test (PRBSTEST) is retrieved.

NAVAILABLE, Not available is retrieved.

Example(s)

* FETC:DATA:TEL:OTN:OPU1:E:PTYP:REC?

Returns the received payload type.

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

Description	<p>This command sets the expected payload signal type to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to PRBStest.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E :PTYPe<wsp>EXPerimental ASYNchronous BISYNch ATM GFP1 VCONcate BSTiming BSNTiming ODUMux RFSTandard RPRopriet NULLtest PRBStest NAVailable</pre>
Parameter(s)	<p>Payload:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: EXPerimental ASYNchronous BISYNch ATM GFP1 VCONcate BSTiming BSNTiming ODUMux RFSTandard RPRopriet NULLtest PRBStest NAVailable.</p>

:SENSe[1..n]:DATA:TELeom:OTN:OPU[1..n]: E:PTYPE

Sets the expected payload signal type to be generated.

EXPerimental, selects the payload type as Experimental.

ASYNchronous, selects the payload type as Asynchronous.

BISYNch, selects the payload type as Bit Synchronous (BISYNch).

ATM, selects the payload type as Asynchronous Transfer Mode (ATM).

GFP1, selects the payload type as Generic Framing Procedure (GFP).

VCONcate, selects the payload type as Virtual Concatenation (VCONcate).

BSTiming, selects the payload type as Bit Stream Timing (BSTiming).

BSNTiming, selects the payload type as Bit Stream No Timing (BSNTiming).

ODUMux, selects the payload type as Optical Data Unit Mux (ODUMux).

RFSTandard, selects the payload type as Reserved Future Standardization (RFSTandard).

RPRopriet, selects the payload type as Reserved Proprietary (RPRopriet).

NULLtest, selects the payload type as NULL Test.

PRBStest, selects the payload type as Pseudo Random Bit Sequence Test (PRBStest).

NAVailable, selects the payload type as Not Available.

**:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:
E:PTYPE**

Example(s)

* SENS:DATA:TEL:OTN:OPU1:E:PTYP EXP

* SENS:DATA:TEL:OTN:OPU1:E:PTYP?

Returns EXPERIMENTAL

See Also

* SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:E:
PTYPE?

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

Description	<p>This query returns the expected payload signal type to be generated for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to PRBStest.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PTYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Payload></p>
Response(s)	<p>Payload:</p> <p>The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the injected payload signal type to be generated.</p> <p>EXPERIMENTAL, Experimental is selected as payload type.</p> <p>ASYNCHRONOUS, Asynchronous is selected as payload type.</p> <p>BISYNCH, Bit Synchronous (BITS) is selected as payload type.</p>

:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]: E:PTYPE?

ATM, Asynchronous Transfer Mode (ATM) is selected as payload type.

GFP, Generic Framing Procedure (GFP) is selected as payload type.

VCONCATENATE, Virtual Concatenation (VCONCATENATE) is selected as payload type.

BSTIMING, Bit Stream Timing (BSTIMING) is selected as payload type.

BSNTIMING, Bit Stream No Timing (BSNTIMING) is selected as payload type.

ODUMUX, Optical Data Unit (ODU) Mux is selected as payload type.

RFSTANDARD, Reserved Future Standardization (RFSTANDARD) is selected as payload type.

RPROPRIET, Reserved Proprietary (RPROPRIET) is selected as payload type.

NULLTEST, NULL Test is selected as payload type.

PRBSTEST, Pseudo Random Bit Sequence Test (PRBSTEST) is selected as payload type.

NAVAILABLE, Not available is selected as payload type.

Example(s)

* SENS:DATA:TEL:OTN:OPU1:E:PTYP EXP

* SENS:DATA:TEL:OTN:OPU1:E:PTYP?

Returns EXPERIMENTAL

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

See Also

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

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

Description This command enables or disables the Payload Type Overwrite feature for non standard rates OTU1e/2e.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
E:PTYPe:OVERwrite:ENABLEd <wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPe:OVERwrite:ENABled

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Payload Type Overwrite feature.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OPU1:E:PTYP:OVER: ENAB ON</p> <p>* SOUR:DATA:TEL:OTN:OPU1:E:PTYP:OVER: ENAB? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPe:OVERwrite:ENABled?</p>

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

Description	This query returns the status of the Payload Type Overwrite feature for non standard rates OTU1e/2e. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PTYPe:OVERwrite:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>

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

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of the Payload Type Overwrite feature.

Example(s)

* SOUR:DATA:TEL:OTN:OPU1:E:PTYP:OVER:
ENAB ON

* SOUR:DATA:TEL:OTN:OPU1:E:PTYP:OVER:
ENAB? Returns 1

See Also

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

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

Description	<p>This command sets the corresponding injected payload type as hexadecimal code for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to #H03.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E :PCODE<wsp><Code></pre>
Parameter(s)	<p>Code:</p> <p>The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the corresponding injected payload type.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OPU1:E:PCOD #H00 * SOUR:DATA:TEL:OTN:OPU1:E:PCOD? Returns #H00</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PCODE?</pre>

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

Description This query returns the corresponding injected payload type as hexadecimal code for non standard rates OTU1e/2e.

At *RST, this value is set to #H03.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
E:PCODE?

Parameter(s) None

Response Syntax <Code>

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

Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the corresponding injected payload type as hexadecimal code.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OPU1:E:PCOD #H00</p> <p>* SOUR:DATA:TEL:OTN:OPU1:E:PCOD?</p> <p>Returns #H00</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: E:PCODE</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
E:PCODE:RECeived?**

Description	<p>This query returns the corresponding received payload type as hexadecimal code for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E: PCODE:RECeived?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Code></p>
Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the received payload code.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OPU1:E:PCOD:REC?</p> <p>Returns the received payload code.</p>

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

Description	<p>This command sets the corresponding expected payload type as hexadecimal code for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to #H03.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E: PCODE<wsp><Code></pre>
Parameter(s)	<p>Code:</p> <p>The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the corresponding expected payload type as hexadecimal code.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:OPU1:E:PCOD #H00 * SENS:DATA:TEL:OTN:OPU1:E:PCOD? Returns #H00</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E: PCODE?</pre>

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

Description	This query returns the corresponding expected payload type as hexadecimal code for non standard rates OTU1e/2e. At *RST, this value is set to #H03.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PCODE?
Parameter(s)	None
Response Syntax	<Code>

**:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:
E:PCODE?**

Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the corresponding expected payload type as hexadecimal code.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OPU1:E:PCOD #H00</p> <p>* SENS:DATA:TEL:OTN:OPU1:E:PCOD?</p> <p>Returns #H00</p>
See Also	<p>* SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:E:PCODE</p>

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

Description	<p>This command enables or disables the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis for non standard rates OTU1e/2e.</p> <p>At *RST this value is set to OFF.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E :PLM<wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OPU1:E:PLM ON</p> <p>* SENS:DATA:TEL:OTN:OPU1:E:PLM? Returns 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PLM?</p>

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

Description	This query returns the status of the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis for non standard rates OTU1e/2e. At *RST this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:E:PLM?
Parameter(s)	None
Response Syntax	<Set>

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

Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.
Example(s)	* SENS:DATA:TEL:OTN:OPU1:E:PLM ON * SENS:DATA:TEL:OTN:OPU1:E:PLM? Returns 1
See Also	* SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:E:PLM

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

Description	<p>This command enables or disables the OPU-MSIM (Multiplex Structure Identifier Mismatch).</p> <p>At *RST this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: MSIM<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the OPU-MSIM (Multiplex Structure Identifier Mismatch)alarm analysis.</p>
Example(s)	<pre>SENS:DATA:TEL:OTN:OPU:MSIM 1</pre>
See Also	<pre>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:PL M?</pre>

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

Description	<p>This query returns the status of the OPU-MSIM (Multiplex Structure Identifier Mismatch).</p> <p>At *RST this value is set to on.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: MSIM?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set:</p> <p>The response data syntax for <Code> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the OPU-MSIM (Multiplex Structure Identifier Mismatch) alarm analysis.</p>
Example(s)	<p>SENS:DATA:TEL:OTN:OPU:MSIM 1 SENS:DATA:TEL:OTN:OPU:MSIM? Returns 1</p>
See Also	<p>SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: PLM</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:TYPE

Description	<p>This command selects the Optical Transport Unit (OTU) alarm type for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to OAIS.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:TYPE<wsp>OAIS OBDI LOF2 OOF1 LOM OOM OBlae OIAE</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OAIS OBDI LOF2 OOF1 LOM OOM OBlae OIAE.</p> <p>Selects the Optical Transport Unit (OTU) alarm type.</p> <p>OAIS, selects 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>

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:TYPE

OBDI, selects 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.

LOF2, selects Loss of Frame (LOF) which generates the errors in all the FAS bits continuously.

OOF1, selects Out of Frame (OOF) which generates the errors in all the FAS bits for 5 consecutive OTU frames repetitively.

LOM, selects Loss of Multiframe (LOM) which generates the errors in multiframe numbers for all the OTU frames continuously.

OOM, selects Out of Multiframe (OOM) which generates the errors in multiframe numbers for 5 consecutive OTU frames repetitively.

OBlae, selects 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.

OIAE, selects 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.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OTU1:E:TYPE OAIS

* SOUR:DATA:TEL:OTN:ALAR:OTU1:E:TYPE?
Returns OAIS

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:TYPE**

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:TYPE?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:TYPE?**

Description	<p>This query returns the Optical Transport Unit (OTU) alarm type for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to O AIS.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Alarm></p>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the Optical Transport Unit (OTU) alarm type.</p> <p>O AIS, 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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:TYPE?**

LOF2, Loss of Frame (LOF) is selected as Optical Transport Unit (OTU) alarm.

OOF1, Out of Frame (OOF) is selected as Optical Transport Unit (OTU) alarm.

LOM, Loss of Multiframe (LOM) is selected as Optical Transport Unit (OTU) alarm.

OOM, Out of Multiframe (OOM) is selected as Optical Transport Unit (OTU) alarm.

OBAE, Optical Transport Unit - Backward Incoming Alignment Error (OTU-BAIE) is selected as Optical Transport Unit (OTU) alarm.

OIAE, Optical Transport Unit - Incoming Alignment Error (OTU-AIE) is selected as Optical Transport Unit (OTU) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OTU1:E:TYPE OAIS

* SOUR:DATA:TEL:OTN:ALAR:OTU1:E:TYPE?
Returns OAIS

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:TYPE

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E**

Description This command enables or disables the status of the Optical Transport Unit (OTU) alarm generation for non standard rates OTU1e/2e.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E<wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E**

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Optical Transport Unit (OTU) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OTU1:E:TYPE OAIS* SOUR:DATA:TEL:OTN:ALAR:OTU1:E ON* SOUR:DATA:TEL:OTN:ALAR:OTU1:E? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E?

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E?

Description	This query returns the status of Optical Transport Unit (OTU) alarm generation for non standard rates OTU1e/2e. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E?**

Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of Optical Transport Unit (OTU) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OTU1:E:TYPE OAIS* SOUR:DATA:TEL:OTN:ALAR:OTU1:E ON* SOUR:DATA:TEL:OTN:ALAR:OTU1:E? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:E

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:MANual:TYPE

Description

This command selects the manual type Optical Transport Unit (OTU) error for non standard rates OTU1e/2e.

At *RST, this value is set to OBIP8.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:MANual:TYPE<wsp>OBIP8|OBEI|
FAS1|MFAS

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:MANual:TYPE

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) type error.</p> <p>OBIP8, selects the ODU - Bit Interleave Parity-8 (OTU-BIP-8) as error.</p> <p>OBEI, selects the ODU - Backward Error Indication (OTU-BEI) as error.</p> <p>FAS1, selects the Frame Alignment Signal (FAS) as error.</p> <p>MFAS, selects the Multiframe Alignment Signal (MFAS) as error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE? Returns OBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:MANual:TYPE?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:MANual:TYPE?

Description This query returns the manual type Optical Transport Unit (OTU) error for non standard rates OTU1e/2e.

At *RST, this value is set to OBIP8.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:MANual:TYPE?

Parameter(s) None

Response Syntax <Error>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:MANual:TYPE?****Response(s)**

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Optical Transport Unit (OTU) type error.

OBIP8, ODU - Bit Interleave Parity-8 (OUT-BIP8) is selected as Optical Transport Unit (OTU) error.

OBEI, ODU - Backward Error Indication (OBE) is selected as Optical Transport Unit (OTU) error.

FAS1, Frame Alignment Signal (FAS) is selected as Optical Transport Unit (OTU) error.

MFAS, Multiframe Alignment Signal (MFAS) is selected as Optical Transport Unit (OTU) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE?
Returns OBIP8

See Also

* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AMOUNT**

Description

This command sets the amount of Optical Transport Unit (OTU) error to be injected for non standard rates OTU1e/2e.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AMOUNT <wsp> <Amount>
|MAXimum | MINimum

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AMOut

Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the amount of Optical Transport Unit (OTU) error. Choices are 1 through 50.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AMOut?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AMOUnt?**

Description	<p>This query returns the amount of Optical Transport Unit (OTU) error injected for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AMOUnt?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<p><Amount></p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AMOut?**

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Optical Transport Unit (OTU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AMOut</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:INJect

Description	<p>This command injects the Optical Transport Unit (OTU) error type for non standard rates OTU1e/2e.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:INJect</pre>
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AMO 15* SOUR:DATA:TEL:OTN:ERR:OTU1:E:INJ
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AMOUNT

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:AUTomated:TYPE****Description**

This command selects the Optical Transport Unit (OTU) error type for automated injection.

At *RST, this value is set to OBIP8.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:AUTomated:TYPE<wsp>OBIP8|
OBEI|FAS1|MFAS

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:AUTomated:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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, selects the ODU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI, selects the ODU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS1, selects the Frame Alignment Signal (FAS) error.</p> <p>MFAS, selects the Multiframe Alignment Signal (MFAS) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:TYPE? Returns OBIP8</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:AUTomated:TYPE****See Also**

* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:AUTomated:TYPE?

* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:AUTomated:RATE

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

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:TYPE?**

Description This query returns the Optical Transport Unit (OTU) error type for automated injection.

At *RST, this value is set to OBIP8.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:TYPE?

Parameter(s) None

Response Syntax <Error>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:TYPE?

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the Optical Transport Unit (OTU) type error for automated injection.</p> <p>OBIP8, ODU - Bit Interleave Parity-8 (OUT-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>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:TYPE? Returns OBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:RATE**

Description

This command sets the injection rate for the selected Optical Transport Unit (OTU) error for non standard rates OTU1e/2e.

At *RST, this value is set to 6.5E-05.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:RATE<wsp> <Rate>
|MAXimum|MINimum

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:RATE

Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected Optical Transport Unit (OTU) error. Choices are 1.0E-09 through 6.5E-05.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:RATE 1.0E-09 * SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:RATE? Returns 1.0E-09</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:RATE? * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Optical Transport Unit (OTU) error for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to 6.5E-05.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:RATE?[<wsp> MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current injected rate will be returned.</p>
Response Syntax	<p><Rate></p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Optical Transport Unit (OTU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:RATE? Returns 1.0E-09</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated

Description

This command enables or disables the selected automated Optical Transport Unit (OTU) error at the rate specified or continuously for non standard rates OTU1e/2e.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated<wsp><Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated****Parameter(s)**

Set:

The program data syntax for <Set> is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability.

ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the automated Optical Transport Unit (OTU) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:RATE
1.0E-09

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated?**

Description	This query returns the status of automated Optical Transport Unit (OTU) error injection for non standard rates OTU1e/2e. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of automated Optical Transport Unit (OTU) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:RATE
1.0E-09

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:CONTInuous

Description

This command enables or disables the automated Optical Transport Unit (OTU) error rate injection continuously.

At *RST, this value is set to OFF.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:  
OTU[1..n]:E:AUTomated:CONTInuous <wsp>  
<Set>
```

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:CONTInuous

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the automated Optical Transport Unit (OTU) error rate injection continuously.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:CONT ON * SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:CONT? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:CONTInuous?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:CONTInuous?**

Description This query returns the status of the automated Optical Transport Unit (OTU) error rate injection continuously.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:CONTInuous?

Parameter(s) None

Response Syntax <Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AUTomated:CONTInuous?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of the automated Optical Transport Unit (OTU) error rate injection continuously.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:CONT ON * SOUR:DATA:TEL:OTN:ERR:OTU1:E:AUT:CONT? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:AUTomated:CONTInuous

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated:TYPE**

Description	<p>This command selects the Forward Error Correction (FEC) error type for automated injection.</p> <p>At *RST, this value is set to FCCW.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:E:FEC:AUTomated:TYPE<wsp> FCCW FUCW FCSYmb FCBit FSCW</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit FSCW.</p> <p>Selects the Forward Error Correction (FEC) error type for automated injection.</p> <p>FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p>

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:E:FEC:AUTomated:TYPE

FUCW, selects the Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.

FCSymb, selects the Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) which generates 1 symbol (byte) containing 8 bits in error.

FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.

FSCW, selects the Forward Error Correction - Stress - Codeword (FEC-STRESS-CW) which generates correctable errors composed of a random number of symbol errors (less or equal to 8) containing a random number of bits distributed all over the OTU frame.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
TYPE? Returns FCCW

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:TYPE**

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:TYPE?

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated:TYPE?**

Description	<p>This query returns the Forward Error Correction (FEC) error type for automated injection.</p> <p>At *RST, this value is set to FCCW.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:E:FEC:AUTomated:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> 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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:TYPE?**

FCBIT, Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) is selected.

FSCW, Forward Error Correction - Stress - Codeword (FEC-STRESS-CW) is selected.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
TYPE? Returns FCCW

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated:RATE****Description**

This command sets the injection rate for the selected Forward Error Correction (FEC) error.

At *RST, this value is set to 1.5E-02.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated:RATE<wsp>
<Rate> | MAXimum | MINimum

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:RATE**

Parameter(s)

Rate:

The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected Forward Error Correction (FEC) error. Choices are 1.0E-07 through 1.5E-02.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
RATE 1.0E-09
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
RATE? Returns 1.0E-09

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:RATE?
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Forward Error Correction (FEC) error.</p> <p>At *RST, this value is set to 1.5E-02.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:E:FEC:AUTomated:RATE?[<wsp> MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current injected rate will be returned.</p>
Response Syntax	<pre><Rate></pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Forward Error Correction (FEC) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT: TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT: RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT: RATE? Returns 1.0E-09</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated****Description**

This command enables or disables the selected automated Forward Error Correction (FEC) error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated**

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the automated Forward Error Correction (FEC) error injection.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT: TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT: RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT ON</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated?</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated?**

Description	This query returns the status of the automated Forward Error Correction (FEC) error injection. At *RST, this value is set to OFF.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:E:FEC:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of the automated Forward Error Correction (FEC) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
RATE 1.0E-09

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT
ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT?
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated:CONTInuous****Description**

This command enables or disables the rate of automated Forward Error Correction (FEC) error injection continuously.

At *RST, this value is set to OFF.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated:CONTInuous
<wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:CONTInuous

Parameter(s)

Set:

The program data syntax for <Set> is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability.

ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the automated Forward Error Correction (FEC) error rate injection continuously.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
CONT ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
CONT? Returns 1

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT
ON

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:CONTInuous?

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AUTomated:CONTInuous?**

Description	This query returns the status of the automated Forward Error Correction (FEC) error rate injection. At *RST, this value is set to OFF.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:E:FEC:AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELecom:OTN:ERRor: OTU[1..n]:E:FEC:AUTomated:CONTInuous?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of the automated Forward Error Correction (FEC) error rate injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
CONT ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT:
CONT? Returns 1

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AUT
ON

See Also

* SOURce[1..n]:DATA:TELecom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:TYPE

* SOURce[1..n]:DATA:TELecom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated

* SOURce[1..n]:DATA:TELecom:OTN:ERRor:
OTU[1..n]:E:FEC:AUTomated:CONTInuous

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:E:TYPE****Description**

This command selects the Optical Payload Unit (OPU) alarm type for non standard rates OTU1e/2e.

At *RST, this value is set to OMSim.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:E:TYPE<wsp>OMSim

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:E:TYPE**

Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: OMSim.</p> <p>Selects the Optical Payload Unit (OPU) alarm type.</p> <p>OMSim, selects the Optical Payload Unit-Multiplex Structure Identifier Mismatch (OPU-MSIM).</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OMS</p> <p>* SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE?</p> <p>Returns OMSIM</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:TYPE?</p>

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

Description	This query returns the Optical Payload Unit (OPU) alarm type for non standard rates OTU1e/2e. At *RST, this value is set to OMSim.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:TYPE?

Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the Optical Payload Unit (OPU) alarm type.</p> <p>OMSIM, OMSIM is selected as Optical Payload Unit (OPU) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OMS</p> <p>* SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE?</p> <p>Returns OMSIM</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:E****Description**

This command enables or disables the Optical Payload Unit (OPU) alarm generation for non standard rates OTU1e/2e.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:E<wsp><Set>

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

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Optical Payload Unit (OPU) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1:E ON* SOUR:DATA:TEL:OTN:ALAR:OPU1:E? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E?

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

Description	This query returns the status of the Optical Payload Unit (OPU) alarm generation for non standard rates OTU1e/2e. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E?
Parameter(s)	None
Response Syntax	<Set>

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

Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the Optical Payload Unit (OPU) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1:E ON* SOUR:DATA:TEL:OTN:ALAR:OPU1:E? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TYPE

Description	<p>This command selects the Optical Data Unit (ODU) alarm type for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to OAIS.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TYPE<wsp>OAIS OBDI OLCK OOCl OFSF OBSF OFSD OBSD LOFLom</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OAIS OBDI OLCK OOCl OFSF OBSF OFSD OBSD LOFLom.</p> <p>Selects the ODU (Optical Data Unit) alarm type. OAIS, selects 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>

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TYPE

OBDI, selects 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).

OLCK, selects 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).

OOCI, selects ODU - Backward Defect Indication (ODU-BDI) which generates a "1" in the BDI (byte 3, bit 5) of the PM overhead field continuously.

OFSF, selects ODU - Forward Signal Fail (ODU-FSF) which generates a "00000001" pattern in the FTFL Byte 0 continuously.

OBSF, selects ODU - Backward Signal Fail (ODU-BSF) which generates a "00000001" pattern in the FTFL Byte 128 continuously.

OFSF, selects ODU - Forward Signal Degrade (ODU-FSD) which generates a "00000010" pattern in the FTFL Byte 0 continuously.

OBSF, selects ODU - Backward Signal Degrade (ODU-BSD) which generates a "00000010" pattern in the FTFL Byte 128 continuously.

LOFLom, selects ODU-Loss of Frame Loss of Multiframe (ODU-LOFLOM) which generate error continuously in FAS and MFAS of a multiplexed test case.

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TYPE****Example(s)**

* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TYPE
OFSF

* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TYPE?
Returns OFSF

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TYPE?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TYPE?**

Description	<p>This query returns the Optical Data Unit (ODU) alarm type for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to OAIS.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Alarm></p>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Message> is defined as a <CHARACTER RESPONSE DATA> 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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TYPE?**

OLCK, ODU - Locked (ODU-LCK) is selected as Optical Data Unit (ODU) alarm.

OOCI, ODU - Open Connection Indication (ODU-OCI) is selected as Optical Data Unit (ODU) alarm.

OFSF, ODU - Forward Signal Fail (ODU-FSF) is selected as Optical Data Unit (ODU) alarm.

OBSF, ODU - Backward Signal Fail (ODU-BSF) is selected as Optical Data Unit (ODU) alarm.

OFSD, ODU - Forward Signal Degrade (ODU-FSD) is selected as Optical Data Unit (ODU) alarm.

OBSD, ODU - Backward Signal Degrade (ODU-BSD) is selected as Optical Data Unit (ODU) alarm.

LOFLOM, ODU Loss of Frame Loss of Multiframe (ODU-LOFLOM) is selected as Optical Data Unit (ODU) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TYPE
OFSF

* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TYPE?
Returns OFSF

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TYPE

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E

Description This command enables or disables the Optical Channel Data Unit (ODU) alarm generation for non standard rates OTU1e/2e.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E<wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E****Parameter(s)****Set:**

The program data syntax for <Set> is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability.

ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the Optical Data Unit (ODU) alarm generation.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TYPE
OFSF

* SOUR:DATA:TEL:OTN:ALAR:ODU1:E ON

* SOUR:DATA:TEL:OTN:ALAR:ODU1:E? Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E?

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E?

Description	This query returns the status of the Optical Data Unit (ODU) alarm generation for non standard rates OTU1e/2e. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of Optical Data Unit (ODU) alarm generation.
Example(s)	* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TYPE OFSF * SOUR:DATA:TEL:OTN:ALAR:ODU1:E ON * SOUR:DATA:TEL:OTN:ALAR:ODU1:E? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:MANual:TYPE**

Description

This command selects the manual type Optical Data Unit (ODU) error for non standard rates OTU1e/2e.

At *RST, this value is set to OBIP8.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:MANual:TYPE<wsp>OBIP8|OBEI

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:MANual:TYPE

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the manual type Optical Data Unit (ODU) error.</p> <p>OBIP8, selects 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, selects 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>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE? Returns OBIP8</p>
See Also	<p>* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:MANual:TYPE?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:MANual:TYPE?

Description	This query returns the manual type Optical Data Unit (ODU) error for non standard rates OTU1e/2e. At *RST, this value is set to OBIP8.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:MANual:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE? Returns OBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:MANual:TYPE</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AMOUNT

Description

This command sets the amount of Optical Data Unit (ODU) error to be injected for non standard rates OTU1e/2e.

At *RST, this value is set to 1.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:  
ODU[1..n]:E:AMOUNT <wsp> <Amount>  
|MAXimum |MINimum
```

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AMOUNT**

Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the amount of Optical Data Unit (ODU) error. Choices are 1 through 50.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AMOUNT?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AMOUnt?

Description	<p>This query returns the amount of Optical Channel Data Unit (ODU) error injected for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AMOUnt?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<p><Amount></p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AMOUNT?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of Optical Data Unit (ODU) error.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AMO 15 * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AMO? Returns 15
See Also	* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:MANual:TYPE * SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:AMOUNT

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:INJect

Description	<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>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:INJect</pre>
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AMO 15* SOUR:DATA:TEL:OTN:ERR:ODU1:E:INJ
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AMOUNT

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:MANual:TYPE**

Description	<p>This command selects the manual type Forward Error Correction (FEC) error for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to FCCW.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:E:FEC:MANual:TYPE<wsp>FCCW FUCW FCSYmb FCBit FSCW</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit FSCW.</p> <p>Selects the Forward Error Correction (FEC) error type.</p> <p>FCCW, selects Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:MANual:TYPE

FUCW, selects Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.

FCSymb, selects Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) which generates 1 symbol (byte) containing 8 bits in error.

FCBit, selects Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.

FSCW, selects Forward Error Correction - Stress - Codeword (FEC-STRESS-CW) which generates correctable errors composed of a random number of symbol errors (less or equal to 8) containing a random number of bits distributed all over the OTU frame.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN:TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN:TYPE? Returns FCCW

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:MANual:TYPE?

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:E:FEC:MANual:TYPE?

Description	<p>This query returns the manual type Forward Error Correction (FEC) error for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to FCCW.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:E:FEC:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the Forward Error Correction (FEC) error type.</p> <p>FCCW, Forward Error Correction - Correctable - Codeword (FEC-CORR-CW) is selected as Forward Error Correction (FEC) error.</p> <p>FUCW, Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) is selected as Forward Error Correction (FEC) error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:MANual:TYPE?**

FCSYMB, Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) is selected as Forward Error Correction (FEC) error.

FCBIT, Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) is selected as Forward Error Correction (FEC) error.

FSCW, Forward Error Correction - Stress - Codeword (FEC-STRESS-CW) is selected as Forward Error Correction (FEC) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN:TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN:TYPE? Returns FCCW

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:MANual:TYPE

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AMOUNT****Description**

This command sets the amount of Forward Error Correction (FEC) error to be injected for non standard rates OTU1e/2e.

At *RST, this value is set to 1.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AMOUNT <wsp> <Amount>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AMOUNT**

Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the amount of Forward Error Correction (FEC) error. Choices are 1 through 50.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN:TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:AMOUNT?</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:E:FEC:AMOUNT?**

Description	<p>This query returns the amount of Forward Error Correction (FEC) error injected for non standard rates OTU1e/2e.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:E:FEC:AMOUNT? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<pre><Amount></pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AMOUNT?

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Forward Error Correction (FEC) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN:TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:AMOUNT</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:INJect**

Description	<p>This command injects the Forward Error Correction (FEC) error type.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN: TYPE FCCW * SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AMO 15 * SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:INJ
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:AMOUNT

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:HISTory?**

Description	<p>This query returns the history status of Optical Transport Unit (OTU) alarm for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:HISTory?<wsp>LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBlae</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBlae.</p> <p>Selects the Optical Transport Unit (OTU) alarm type.</p> <p>LOF2, selects Loss Of Frame (LOF) when OOF is present for at least 3 ms.</p> <p>OOF1, selects 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>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:HISTory?**

LOM, selects Loss Of Multiframe (LOM) when OOM (Out of Multiframe) is present for at least 3 ms.

OOM, selects Out-Of-Multiframe (OOM) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.

OAIS, selects 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.

OTIM, selects 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.

OBDI, selects 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.

OIAE, selects 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.

OBlae, selects 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.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:HISTory?**

Response Syntax <History>

Response(s)

History:

The response data syntax for <History> 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.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OTU1:E:TYPE

OAIS

* SOUR:DATA:TEL:OTN:ALAR:OTU1:E ON

* FETC:DATA:TEL:OTN:ALAR:OTU1:E:HIST? OAIS

Returns the alarm history.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:

OTU[1..n]:E:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:

OTU[1..n]E:

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:SEConds?

Description	<p>This query returns the number of seconds within which Optical Transport Unit (OTU) alarm occurred for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:SEConds?<wsp>LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBlae</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBlae.</p> <p>Selects the Optical Transport Unit (OTU) alarm type.</p> <p>LOF2, selects Loss Of Frame (LOF) when OOF is present for at least 3 ms.</p> <p>OOF1, selects 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>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:SECOndS?

LOM, selects Loss Of Multiframe (LOM) when OOM (Out of Multiframe) is present for at least 3 ms.

OOM, selects Out-Of-Multiframe (OOM) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.

OAIS, selects 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.

OTIM, selects 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.

OBDI, selects 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.

OIAE, selects 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.

OBlae, selects 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.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:SEConds?****Response Syntax** <Seconds>**Response(s)**

Seconds:

The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the number of seconds of Optical Transport Unit (OTU) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OTU1:E:TYPE
OAIS

* SOUR:DATA:TEL:OTN:ALAR:OTU1:E ON

* FETC:DATA:TEL:OTN:ALAR:OTU1:E:SEC? OAIS

Returns the number of seconds of OTU alarm.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:CURRent?**

Description	<p>This query returns the current status of Optical Transport Unit (OTU) alarm for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:E:CURRent?<wsp>LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBlae</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF2 OOF1 LOM OOM OAIS OTIM OBDI OIAE OBlae.</p> <p>Selects the Optical Transport Unit (OTU) alarm type.</p> <p>LOF2, selects Loss Of Frame (LOF) when OOF is present for at least 3 ms.</p> <p>OOF1, selects 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>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:CURREnt?**

LOM, selects Loss Of Multiframe (LOM) when OOM (Out of Multiframe) is present for at least 3 ms.

OOM, selects Out-Of-Multiframe (OOM) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.

OAIS, selects 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.

OTIM, selects 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.

OBDI, selects 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.

OIAE, selects 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.

OBlae, selects 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.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:E:CURRent?**

Response Syntax <Current>

Response(s)

Current:

The response data syntax for <Current> 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.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OTU1:E:TYPE

OAIS

* SOUR:DATA:TEL:OTN:ALAR:OTU1:E ON

* FETC:DATA:TEL:OTN:ALAR:OTU1:E:CURR?

OAIS Returns the current alarm status.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:

OTU[1..n]:E:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:

OTU[1..n]:E

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:HISTory?

Description	<p>This query returns the history status of Optical Transport Unit (OTU) error for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:HISTory?<wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8, selects ODU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI, selects ODU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS1, selects Frame Alignment Signal (FAS) error.</p> <p>MFAS, selects Multiframe Alignment Signal (MFAS) error.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:E:HISTory?**

Response Syntax <History>

Response(s) History:
The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the history status of Optical Transport Unit (OTU) 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.

Example(s) * SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AMO 15
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:E:HIST? OBIP8
Returns the error history.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:E:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:E:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:E:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:SEConds?**

Description	<p>This query returns the number of seconds within which Optical Transport Unit (OTU) error occurred for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:SEConds?<wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8, selects ODU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI, selects ODU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS1, selects Frame Alignment Signal (FAS) error.</p> <p>MFAS, selects Multiframe Alignment Signal (MFAS) error.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:SEConds?**

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Optical Transport Unit (OTU) error.

Example(s) * SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AMO 15
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:E:SEC? OBIP8
Returns the number of errored seconds.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:INJect

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:CURRent?

Description	<p>This query returns the current status of Optical Transport Unit (OTU) error for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:CURRent?<wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8, selects ODU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI, selects ODU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS1, selects Frame Alignment Signal (FAS) error.</p> <p>MFAS, selects Multiframe Alignment Signal (MFAS) error.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:E:CURREnt?**

Response Syntax <Current>

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> > element.

Returns the current status of Optical Transport Unit (OTU) 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:OTN:ERR:OTU1:E:MAN:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AMO 15

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:INJ

* FETC:DATA:TEL:OTN:ERR:OTU1:E:CURR?
OBIP8 Returns the current error status.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:E:MANual:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:E:AMOnt

* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:E:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:COUNT?**

Description	<p>This query returns the count of Optical Transport Unit (OTU) error for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:COUNT? <wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8, selects ODU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI, selects ODU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS1, selects Frame Alignment Signal (FAS) error.</p> <p>MFAS, selects Multiframe Alignment Signal (MFAS) error.</p>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:COUNT?

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of Optical Transport Unit (OTU) error.

Example(s) * SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AMO 15
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:E:COUN?
OBIP8 Returns the error count.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:RATE?**

Description	<p>This query returns the current rate of Optical Transport Unit (OTU) error for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:RATE?<wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8, selects ODU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI, selects ODU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS1, selects Frame Alignment Signal (FAS) error.</p> <p>MFAS, selects Multiframe Alignment Signal (MFAS) error.</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERror:
OTU[1..n]:E:RATE?****Response Syntax** <Rate>**Response(s)**

Rate:

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the current rate of Optical Transport Unit (OTU) error.

Example(s)

```
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:MAN:TYPE  
OBIP8  
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:AMO 15  
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:INJ  
* FETC:DATA:TEL:OTN:ERR:OTU1:E:RATE?  
OBIP8 Returns the current error rate.
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:OTN:ERror:  
OTU[1..n]:E:MANual:TYPE  
* SOURce[1..n]:DATA:TELEcom:OTN:ERror:  
OTU[1..n]:E:AMOUNT  
* SOURce[1..n]:DATA:TELEcom:OTN:ERror:  
OTU[1..n]:E:INJect
```

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:HISTory?

Description

This query returns the history status of Optical Data Unit (ODU) alarm for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:  
ODU[1..n]:E:HISTory?<wsp>OAIS|OBDI|OLCK|  
OOCI|OFSF|OBSF|OTIM|OFSD|OBSD|LOFLom
```

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

```
OAIS|OBDI|OLCK|OOCI|OFSF|OBSF|OTIM|  
OFSD|OBSD|LOFLom.
```

Selects the Optical Data Unit (ODU) alarm type.

OAIS, selects 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.

:FETCh[1..n]:DATA:TELecom:OTN:ALARm: ODU[1..n]:E:HISTory?

OBDI, selects 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.

OLCK, selects ODU - Backward Defect indication (OLCK) indicates that the STAT information detected is "101" for at least 3 consecutive frames.

OOCI, selects ODU - Open Connection Indication (OOCI) indicates that the STAT information detected is "110" for at least 3 consecutive frames.

OFSF, selects ODU - Forward Signal Fail (ODU-FSF) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000001".

OBSF, selects ODU - Backward Signal Fail (ODU-BSF) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000001".

OTIM, selects 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.

**:FETCh[1..n]:DATA:TELeCom:OTN:ALARm:
ODU[1..n]:E:HISTory?**

OFSd, selects ODU - Forward Signal Degrade (ODU-FSD) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000010"

OBSD, selects ODU - Backward Signal Degrade (ODU-BSD) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000010".

LOFLom, selects ODU-Loss of Frame Loss of Multiframe (ODU-LOFLOM) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.

Response Syntax <History>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:HISTory?

Response(s)

History:

The response data syntax for <History> 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.

Example(s)

```
* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TYPE  
O AIS  
* SOUR:DATA:TEL:OTN:ALAR:ODU1:E ON  
* FETC:DATA:TEL:OTN:ALAR:ODU1:E:HIST?  
O AIS Returns the alarm history.
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:  
ODU[1..n]:E:TYPE  
* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:  
ODU[1..n]:E
```

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:SEConds?

Description	<p>This query returns the number of seconds within which Optical Data Unit (ODU) alarm occurred for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:SEConds?<wsp>OAIS OBDI OLCK OOCI OFSF OBSF OTIM OFSD OBSD LOFLom</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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. OAIS, selects 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>

:FETCh[1..n]:DATA:TELecom:OTN:ALARm: ODU[1..n]:E:SECOndS?

OBDI, selects 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.

OLCK, selects ODU - Backward Defect indication (OLCK) indicates that the STAT information detected is "101" for at least 3 consecutive frames.

OOCI, selects ODU - Open Connection Indication (OOCI) indicates that the STAT information detected is "110" for at least 3 consecutive frames.

OFSF, selects ODU - Forward Signal Fail (ODU-FSF) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000001".

OBSF, selects ODU - Backward Signal Fail (ODU-BSF) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000001".

OTIM, selects 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.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:SEConds?**

OFSd, selects ODU - Forward Signal Degrade (ODU-FSD) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000010"

OBSD, selects ODU - Backward Signal Degrade (ODU-BSD) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000010".

LOFLom, selects ODU-Loss of Frame Loss of Multiframe (ODU-LOFLOM) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.

Response Syntax <Seconds>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:SEConds?

Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of seconds of Optical Data Unit (ODU) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TYPE OAIS</p> <p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:E ON</p> <p>* FETC:DATA:TEL:OTN:ALAR:ODU1:E:SEC? OAIS</p> <p>Returns the number of seconds of ODU alarm.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E</p>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:CURRent?

Description	<p>This query returns the current status of Optical Data Unit (ODU) alarm for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:CURRent?<wsp>OAIS OBDI OLCK OOCI OFSF OBSF OTIM OFSD OBSD LOFLom</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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. OAIS, selects 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>

:FETCH[1..n]:DATA:TELEcom:OTN:ALARM: ODU[1..n]:E:CURRENT?

OBDI, selects 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.

OLCK, selects ODU - Backward Defect indication (OLCK) indicates that the STAT information detected is "101" for at least 3 consecutive frames.

OOCI, selects ODU - Open Connection Indication (OOCI) indicates that the STAT information detected is "110" for at least 3 consecutive frames.

OFSF, selects ODU - Forward Signal Fail (ODU-FSF) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000001".

OBSF, selects ODU - Backward Signal Fail (ODU-BSF) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000001".

OTIM, selects 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.

OFSF, selects ODU - Forward Signal Degradation (ODU-FSD) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000010"

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:E:CURRent?**

OBSD, selects ODU - Backward Signal Degrade (ODU-BSD) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000010".

LOFLom, selects ODU-Loss of Frame Loss of Multiframe (ODU-LOFLOM) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.

Response Syntax <Current>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:CURRent?

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:ODU1:E:TYPE O AIS* SOUR:DATA:TEL:OTN:ALAR:ODU1:E ON* FETC:DATA:TEL:OTN:ALAR:ODU1:E:CURR? O AIS Returns the current alarm status.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:E

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:HISTory?**

Description	<p>This query returns the history status of Optical Data Unit (ODU) error for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:E:HISTory?<wsp>OBIP8 OBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8, selects 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, selects 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>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:HISTory?**

Response Syntax <History>

Response(s) History:
The response data syntax for <History> 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.

Example(s) * SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AMO 15
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:INJ
* FETC:DATA:TEL:OTN:ERR:ODU1:E:HIST? OBIP8
Returns the error history.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AMOut
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:SECOnds?**

Description	<p>This query returns the number of seconds within which Optical Data Unit (ODU) error occurred for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:E:SECOnds?<wsp>OBIP8 OBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8, selects 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, selects 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>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:SEConds?**

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Message> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Optical Data Unit (ODU) error.

Example(s) * SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AMO 15
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:INJ
* FETC:DATA:TEL:OTN:ERR:ODU1:E:SEC? OBIP8
Returns the number of errored seconds.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:CURREnt?**

Description	<p>This query returns the current status of Optical Data Unit (ODU) error for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:E:CURREnt?<wsp>OBIP8 OBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8, selects 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, selects 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>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:CURREnt?**

Response Syntax <Current>

Response(s)

Current:

The response data syntax for <Current> 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.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AMO 15

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:INJ

* FETC:DATA:TEL:OTN:ERR:ODU1:E:CURR?
OBIP8 Returns the current error status.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:MANual:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AMOUNT

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:COUNT?**

Description	<p>This query returns the count of Optical Data Unit (ODU) error for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:E:COUNT?<wsp>OBIP8 OBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI.</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8, selects 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, selects 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>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:COUNT?

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of Optical Data Unit (ODU) error.

Example(s) * SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AMO 15
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:INJ
* FETC:DATA:TEL:OTN:ERR:ODU1:E:COUN?
OBIP8 Returns the error count.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:EROR:
ODU[1..n]:E:RATE?****Description**

This query returns the current rate of Optical Data Unit (ODU) error for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:EROR:
ODU[1..n]:E:RATE?<wsp>OBIP8|OBEI

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8|OBEI.

Selects the Optical Data Unit (ODU) error type.

OBIP8, selects 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).

OBEI, selects 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.

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:RATE?

Response Syntax <Rate>

Response(s)

Rate:

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the current rate of Optical Data Unit (ODU) error.

Example(s)

```
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AMO 15
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:INJ
* FETC:DATA:TEL:OTN:ERR:ODU1:E:RATE?
OBIP8 Returns the error rate.
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:INJECT
```


**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TYPE****Description**

This command selects the Optical Data Unit (ODU) error type for automated injection.

At *RST, this value is set to OBIP8.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TYPE<wsp>OBIP8 |
OBEI

:SOURce[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:E:AUTomated:TYPE

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8|OBEI.

Selects the Optical Data Unit (ODU) error type.

OBIP8, selects 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).

OBEI, selects 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.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TYPE

OBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TYPE?

Returns OBIP8

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TYPE**

See Also

* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:TYPE?

* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated:RATE

* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:E:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TYPE?**

Description	This query returns the Optical Data Unit (ODU) error type for automated injection. At *RST, this value is set to OBIP8.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TYPE?****Response(s)**

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Optical Data Unit (ODU) error type for automated injection.

OBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as Optical Data Unit (ODU) error.

OBEI, ODU - Backward Error Indication (ODU-BEI) is selected as Optical Data Unit (ODU) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TYPE?
Returns OBIP8

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:RATE

Description

This command sets the injection rate for the selected Optical Data Unit (ODU) error.

At *RST, this value is device dependent.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:RATE<wsp> <Rate>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:AUTomated:RATE**

Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected Optical Data Unit (ODU) error.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:RATE 1.0E-09 * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:RATE? Returns 1.0E-09</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:E:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:E:AUTomated:RATE? * SOURce[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:E:AUTomated</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:E:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:E:AUTomated:RATE?[<wsp> MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the injected rate will be returned.</p>
Response Syntax	<pre><Rate></pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:RATE?****Response(s)**

Rate:

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the injection rate for the selected Optical Data Unit (ODU) error.

Example(s)

```
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TYPE  
OBIP8
```

```
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:RATE  
1.0E-09
```

```
* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:RATE?  
Returns 1.0E-09
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:  
ODU[1..n]:E:AUTomated:TYPE
```

```
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:  
ODU[1..n]:E:AUTomated:RATE
```

```
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:  
ODU[1..n]:E:AUTomated
```

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated

Description

This command enables or disables the selected automated Optical Data Unit (ODU) error at the rate specified or continuously for non standard rates OTU1e/2e.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated <wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the automated Optical Data Unit (ODU) error injection.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:RATE 1.0E-09 * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT ON * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated?

Description	This query returns the status of the automated Optical Data Unit (ODU) error injection for non standard rates OTU1e/2e. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated?

Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the automated Optical Data Unit (ODU) error injection.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:RATE 1.0E-09 * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT ON * SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:CONTInuous**

Description

This command enables or disables the automated Optical Data Unit (ODU) error injection rate continuously for non standard rates OTU1e/2e.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:CONTInuous <wsp>
<Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:CONTInuous

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the automated Optical Data Unit (ODU) error injection rate.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:CONT ON</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT: CONT? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:AUTomated:CONTInuous?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:CONTInuous?**

Description This query returns the status of the automated Optical Data Unit (ODU) error injection rate for non standard rates OTU1e/2e.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:CONTInuous?

Parameter(s) None

Response Syntax <Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:CONTInuous?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of the automated Optical Data Unit (ODU) error injection rate.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:CONT
ON

* SOUR:DATA:TEL:OTN:ERR:ODU1:E:AUT:
CONT? Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:E:AUTomated:CONTInuous

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:HISTory?

Description This query returns the history status of Optical Payload Unit (OPU) alarm for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:E:HISTory? <wsp>OPLM|OMSim

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPLM|OMSim.
Selects the Optical Payload Unit (OPU) alarm type.
OPLM, selects the Optical Payload Unit Payload Mismatch (OPU-PLM) alarm.
OMSim, selects the Optical Payload Unit Multiplex Structure Identifier Mismatch (OPU-MSIM) alarm.

Response Syntax <History>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:E:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1:E ON* FETC:DATA:TEL:OTN:ALAR:OPU1:E:HIST? OMS <p>Returns the alarm history.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:SEConds?

Description This query returns the number of seconds within which Optical Payload Unit (OPU) alarm occurred for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:E:SEConds?<wsp>OPLM|OMSim

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPLM|OMSim.
Selects the Optical Payload Unit (OPU) alarm type.
OPLM, selects the Optical Payload Unit Payload Mismatch (OPU-PLM) alarm.
OMSim, selects the Optical Payload Unit Multiplex Structure Identifier Mismatch (OPU-MSIM) alarm.

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELeom:OTN:ALARm:
OPU[1..n]:E:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Payload Unit (OPU) alarm.
Example(s)	* SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OMS * SOUR:DATA:TEL:OTN:ALAR:OPU1:E ON * FETC:DATA:TEL:OTN:ALAR:OPU1:E:SEC? OMS Returns the number of seconds of OPU alarm.
See Also	* SOURce[1..n]:DATA:TELeom:OTN:ALARm: OPU[1..n]:E:TYPE * SOURce[1..n]:DATA:TELeom:OTN:ALARm: OPU[1..n]:E

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:CURRent?

Description	<p>This query returns the current status of Optical Payload Unit (OPU) alarm for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:E:CURRent?<wsp>OPLM OMSim</p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPLM OMSim.</p> <p>Selects the Optical Payload Unit (OPU) alarm type.</p> <p>OPLM, selects the Optical Payload Unit Payload Mismatch (OPU-PLM) alarm.</p> <p>OMSim, selects the Optical Payload Unit Multiplex Structure Identifier Mismatch (OPU-MSIM) alarm.</p>
Response Syntax	<p><Current></p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:E:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of 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 is not running.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:E:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1:E ON* FETC:DATA:TEL:OTN:ALAR:OPU1:E:CURR? <p>OMS Returns the current alarm status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:E

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:HISTory?**

Description

This query returns the history status of Forward Error Correction (FEC) error for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:HISTory? <wsp>FCCW|FUCW|
FCSYmb|FCBit

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FCCW|FUCW|FCSYmb|FCBit.

Selects Forward Error Correction (FEC) error type.

FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:HISTory?**

FUCW, selects the Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.

FCSYmb, selects the Forward Error Correction - Correctable - Symbol (FEC-UNCORR-CW) which generates 1 symbol (byte) containing 8 bits in error.

FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.

Response Syntax <History>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:HISTory?**

Response(s) History:
The response data syntax for <History> 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.

Example(s) * SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AMO
15
* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:E:FEC:HIST?
FCCW Returns the error history.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:SEConds?****Description**

This query returns the number of seconds within which Forward Error Correction (FEC) error occurred for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:SEConds?<wsp>FCCW|
FUCW|FCSYmb|FCBit

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FCCW|FUCW|FCSYmb|FCBit.

Selects Forward Error Correction (FEC) error type.

FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.

**:FETCh[1..n]:DATA:TELeCom:OTN:ERROr:
OTU[1..n]:E:FEC:SEConds?**

FUCW, selects the Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.

FCSYmb, selects the Forward Error Correction - Correctable - Symbol (FEC-UNCORR-CW) which generates 1 symbol (byte) containing 8 bits in error.

FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.

Response Syntax <Seconds>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:SEConds?

Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of seconds of Forward Error Correction (FEC) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN:TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:INJ</p> <p>* FETC:DATA:TEL:OTN:ERR:OTU1:E:FEC:SEC? FCCW Returns the number of errored seconds.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:AMOUNT</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:INJect</p>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:CURRent?

Description This query returns the current status of Forward Error Correction (FEC) error for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:CURRent? <wsp>FCCW|
FUCW|FCSYmb|FCBit

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FCCW|FUCW|FCSYmb|FCBit.
Selects Forward Error Correction (FEC) error type.
FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:CURRENT?**

FUCW, selects the Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.

FCSymb, selects the Forward Error Correction - Correctable - Symbol (FEC-UNCORR-CW) which generates 1 symbol (byte) containing 8 bits in error.

FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.

Response Syntax <Current>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:CURREnt?**

Response(s)

Current:

The response data syntax for <Current> 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 is not running.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AMO
15

* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:INJ

* FETC:DATA:TEL:OTN:ERR:OTU1:E:FEC:CURREnt?
FCCW Returns the current error status.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:MANual:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:AMOUNT

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:INJECT

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:COUNT?****Description**

This query returns the count of Forward Error Correction (FEC) error for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:COUNT? <wsp>FCCW|FUCW|
FCSYmb|FCBit

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FCCW|FUCW|FCSYmb|FCBit.

Selects Forward Error Correction (FEC) error type.

FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.

:FETCh[1..n]:DATA:TELecom:OTN:ERRor: OTU[1..n]:E:FEC:COUNT?

FUCW, selects the Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.

FCSYmb, selects the Forward Error Correction - Correctable - Symbol (FEC-UNCORR-CW) which generates 1 symbol (byte) containing 8 bits in error.

FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.

Response Syntax <Count>

:FETCh[1..n]:DATA:TELecom:OTN:ERRor: OTU[1..n]:E:FEC:COUNT?

Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the count of Forward Error Correction (FEC) error.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN: TYPE FCCW * SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AMO 15 * SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:INJ * FETC:DATA:TEL:OTN:ERR:OTU1:E:FEC:COUN? FCCW Returns the error count.</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELecom:OTN:ERRor: OTU[1..n]:E:FEC:MANual:TYPE * SOURce[1..n]:DATA:TELecom:OTN:ERRor: OTU[1..n]:E:FEC:AMOUNT * SOURce[1..n]:DATA:TELecom:OTN:ERRor: OTU[1..n]:E:FEC:INJect</pre>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:E:FEC:RATE?

Description

This query returns the current rate of Forward Error Correction (FEC) error for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:RATE? <wsp>FCCW|FUCW|
FCSYmb|FCBit

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FCCW|FUCW|FCSYmb|FCBit.

Selects Forward Error Correction (FEC) error type.

FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:RATE?**

FUCW, selects the Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) which generates 16 symbol (bytes) containing 8 bits in error each, in each codeword.

FCSymb, selects the Forward Error Correction - Correctable - Symbol (FEC-UNCORR-CW) which generates 1 symbol (byte) containing 8 bits in error.

FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.

Response Syntax <Rate>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:E:FEC:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of Forward Error Correction (FEC) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:MAN:TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:E:FEC:INJ</p> <p>* FETC:DATA:TEL:OTN:ERR:OTU1:E:FEC:RATE? FCCW Returns the current error rate.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:AMOUNT</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:INJect</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:E:OVERhead****Description**

This command sets the overhead byte values for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:  
OTU[1..n]:E: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:OVERhead

Parameter(s)

Overhead:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

OA111|OA112|OA113|OA214|OA215|OA216|MFAS17|SM18|SM19|SM110|GCC0111|GCC0112|RES113|RES114.

Selects overhead bytes for non standard rates OTU1e/2e.

OA111, selects OA1 as overhead byte.

OA112, selects OA1 as overhead byte.

OA113, selects OA1 as overhead byte.

OA214, selects OA2 as overhead byte.

OA215, selects OA2 as overhead byte.

OA216, selects OA2 as overhead byte.

MFAS17, selects MFAS as overhead byte.

SM18, selects SM as overhead byte.

SM19, selects SM as overhead byte.

SM110, selects SM as overhead byte.

GCC0111, selects GCC0 as overhead byte.

GCC0112, selects GCC0 as overhead byte.

RES113, selects RES as overhead byte.

RES114, selects RES as overhead byte.

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.

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:E:OVERhead**

Value:

The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the overhead byte values.

Example(s)

* SOUR:DATA:TEL:OTN:OH:OTU1:E:OVER
OA111, #HF6

* SOUR:DATA:TEL:OTN:OH:OTU1:E:OVER?
OA111 Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:E:OVERhead?

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:OVERhead?

Description

This query returns the overhead byte values for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:  
OTU[1..n]:E:OVERhead?<wsp>OA111|OA112|  
OA113|OA214|OA215|OA216|MFAS17|SM18|  
SM19|SM110|GCC0111|GCC0112|RES113|  
RES114
```

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:OVERhead?

Parameter(s)	Overhead:
	<p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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 OTU1e/2e.</p> <p>OA111, selects OA1 as overhead byte. OA112, selects OA1 as overhead byte. OA113, selects OA1 as overhead byte. OA214, selects OA2 as overhead byte. OA215, selects OA2 as overhead byte. OA216, selects OA2 as overhead byte. MFAS17, selects MFAS as overhead byte. SM18, selects SM as overhead byte. SM19, selects SM as overhead byte. SM110, selects SM as overhead byte. GCC0111, selects GCC0 as overhead byte. GCC0112, selects GCC0 as overhead byte. RES113, selects RES as overhead byte. RES114, selects RES as overhead byte.</p>

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:OVERhead?

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.

Response Syntax <Value>

Response(s) Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the overhead byte values in hexadecimal format.

Example(s)

* SOUR:DATA:TEL:OTN:OH:OTU1:E:OVER
OA111, #HF6

* SOUR:DATA:TEL:OTN:OH:OTU1:E:OVER?
OA111 Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:E:OVERhead

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:E:DEFault**

Description	<p>This command resets or overwrites the overhead byte values for non standard rates OTU1e/2e.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:DEFault
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:OH:OTU1:E:DEF

:SENSe[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:OVERhead?

Description	<p>This query returns the overhead byte values for non standard rates OTU1e/2e.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n] :E:OVERhead?<wsp>OA111 OA112 OA113 OA214 OA215 OA216 MFAS17 SM18 SM19 SM110 GCC0111 GCC0112 RES113 RES114</pre>
Parameter(s)	<p>Overhead:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OA111 OA112 OA113 OA214 OA215 OA216 MFAS17 SM18 SM19 SM110 GCC0111 GCC0112 RES113 RES114.</p>

**:SENSe[1..n]:DATA:TELeCom:OTN:OH:
OTU[1..n]:E:OVERhead?**

Selects overhead bytes for non standard rates OTU1e/2e.

OA111, selects OA1 as overhead byte.

OA112, selects OA1 as overhead byte.

OA113, selects OA1 as overhead byte.

OA214, selects OA2 as overhead byte.

OA215, selects OA2 as overhead byte.

OA216, selects OA2 as overhead byte.

MFAS17, selects MFAS as overhead byte.

SM18, selects SM as overhead byte.

SM19, selects SM as overhead byte.

SM110, selects SM as overhead byte.

GCC0111, selects GCC0 as overhead byte.

GCC0112, selects GCC0 as overhead byte.

RES113, selects RES as overhead byte.

RES114, selects RES as overhead byte.

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.

Response Syntax <Value>

:SENSe[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:OVERhead?

Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the overhead byte values in hexadecimal format.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OH:OTU1:E:OVER? OA111</p> <p>Returns the OTU overhead byte values.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:E:OVERhead</p>

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:E:OVERhead

Description This command sets the overhead byte values for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:E:OVERhead<wsp>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,
<Value>

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[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 <CHARACTER PROGRAM DATA> 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.

Selects overhead bytes for non standard rates OTU1e/2e.

RES21, selects RES as overhead byte.

RES22, selects RES as overhead byte.

RES23, selects RES as overhead byte.

TCMACT24, selects TCM as overhead byte.

TCM625, selects TCM6 as overhead byte.

TCM626, selects TCM6 as overhead byte.

TCM627, selects TCM6 as overhead byte.

TCM528, selects TCM5 as overhead byte.

TCM529, selects TCM5 as overhead byte.

TCM5210, selects TCM5 as overhead byte.

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:E:OVERhead**

TCM4211, selects TCM4 as overhead byte.

TCM4212, selects TCM4 as overhead byte.

TCM4213, selects TCM4 as overhead byte.

FTFL214, selects FTFL as overhead byte.

TCMACT331, selects TCM3 as overhead byte.

TCM332, selects TCM3 as overhead byte.

TCM333, selects TCM3 as overhead byte.

TCM234, selects TCM2 as overhead byte.

TCM235, selects TCM2 as overhead byte.

TCM236, selects TCM2 as overhead byte.

TCM137, selects TCM1 as overhead byte.

TCM138, selects TCM1 as overhead byte.

TCM139, selects TCM1 as overhead byte.

PM310, selects PM as overhead byte.

PM311, selects PM as overhead byte.

PM312, selects PM as overhead byte.

EXP313, selects EXP as overhead byte.

EXP314, selects EXP as overhead byte.

GCC141, selects GCC1 as overhead byte.

GCC142, selects GCC1 as overhead byte.

GCC243, selects GCC2 as overhead byte.

GCC244, selects GCC2 as overhead byte.

APSPCC45, selects APSPCC as overhead byte.

APSPCC46, selects APSPCC as overhead byte.

APSPCC47, selects APSPCC as overhead byte.

APSPCC48, selects APSPCC as overhead byte.

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:E:OVERhead**

RES49, selects RES as overhead byte.
RES410, selects RES as overhead byte.
RES411, selects RES as overhead byte.
RES412, selects RES as overhead byte.
RES413, selects RES as overhead byte.
RES414 selects RES as overhead byte.

Value:

The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the overhead byte values.

Example(s)

* SOUR:DATA:TEL:OTN:OH:ODU1:E:OVER
RES21, #HF6

* SOUR:DATA:TEL:OTN:OH:ODU1:E:OVER?
RES21 Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:E:OVERhead?

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:E:OVERhead?

Description

This query returns the overhead byte values for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:E:OVERhead? <wsp>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
```

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[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 <CHARACTER PROGRAM DATA> 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.

Selects overhead bytes for non standard rates OTU1e/2e.

RES21, selects RES as overhead byte.

RES22, selects RES as overhead byte.

RES23, selects RES as overhead byte.

TCMACT24, selects TCM as overhead byte.

TCM625, selects TCM6 as overhead byte.

TCM626, selects TCM6 as overhead byte.

TCM627, selects TCM6 as overhead byte.

TCM528, selects TCM5 as overhead byte.

TCM529, selects TCM5 as overhead byte.

TCM5210, selects TCM5 as overhead byte.

**:SOURCE[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:E:OVERhead?**

TCM4211, selects TCM4 as overhead byte.
TCM4212, selects TCM4 as overhead byte.
TCM4213, selects TCM4 as overhead byte.
FTFL214, selects FTFL as overhead byte.
TCM331, selects TCM3 as overhead byte.
TCM332, selects TCM3 as overhead byte.
TCM333, selects TCM3 as overhead byte.
TCM234, selects TCM2 as overhead byte.
TCM235, selects TCM2 as overhead byte.
TCM236, selects TCM2 as overhead byte.
TCM137, selects TCM1 as overhead byte.
TCM138, selects TCM1 as overhead byte.
TCM139, selects TCM1 as overhead byte.
PM310, selects PM as overhead byte.
PM311, selects PM as overhead byte.
PM312, selects PM as overhead byte.
EXP313, selects EXP as overhead byte.
EXP314, selects EXP as overhead byte.
GCC141, selects GCC1 as overhead byte.
GCC142, selects GCC1 as overhead byte.
GCC243, selects GCC2 as overhead byte.
GCC244, selects GCC2 as overhead byte.
APSPCC45, selects APSPCC as overhead byte.
APSPCC46, selects APSPCC as overhead byte.
APSPCC47, selects APSPCC as overhead byte.

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:E:OVERhead?

APSPCC48, selects APSPCC as overhead byte.

RES49, selects RES as overhead byte.

RES410, selects RES as overhead byte.

RES411, selects RES as overhead byte.

RES412, selects RES as overhead byte.

RES413, selects RES as overhead byte.

RES414, selects RES as overhead byte.

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.

Response Syntax <Value>

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:E:OVERhead?**

Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the overhead byte values in hexadecimal format.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OH:ODU1:E:OVER RES21, #HF6</p> <p>* SOUR:DATA:TEL:OTN:OH:ODU1:E:OVER? RES21 Returns #HF6</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:E:OVERhead</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:E:DEFault**

Description	<p>This command resets or overwrites the overhead byte values for non standard rates OTU1e/2e.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:E:DEFault</p>
Parameter(s)	<p>None</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OH:ODU1:E:DEF</p>

:SENSe[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:E:OVERhead?

Description

This query returns the overhead byte values for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

```
:SENSe[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]
:E:OVERhead?<wsp>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
```

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:
ODU[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 <CHARACTER PROGRAM DATA> 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 | RES41
0 | RES411 | RES412 | RES413 | RES414.

Selects overhead bytes.

RES21, selects RES as overhead byte.

RES22, selects RES as overhead byte.

RES23, selects RES as overhead byte.

TCMACT24, selects TCM as overhead byte.

TCM625, selects TCM6 as overhead byte.

TCM626, selects TCM6 as overhead byte.

TCM627, selects TCM6 as overhead byte.

TCM528, selects TCM5 as overhead byte.

TCM529, selects TCM5 as overhead byte.

TCM5210, selects TCM5 as overhead byte.

**:SENSe[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:E:OVERhead?**

TCM4211, selects TCM4 as overhead byte.
TCM4212, selects TCM4 as overhead byte.
TCM4213, selects TCM4 as overhead byte.
FTFL214, selects FTFL as overhead byte.
TCM331, selects TCM3 as overhead byte.
TCM332, selects TCM3 as overhead byte.
TCM333, selects TCM3 as overhead byte.
TCM234, selects TCM2 as overhead byte.
TCM235, selects TCM2 as overhead byte.
TCM236, selects TCM2 as overhead byte.
TCM137, selects TCM1 as overhead byte.
TCM138, selects TCM1 as overhead byte.
TCM139, selects TCM1 as overhead byte.
PM310, selects PM as overhead byte.
PM311, selects PM as overhead byte.
PM312, selects PM as overhead byte.
EXP313, selects EXP as overhead byte.
EXP314, selects EXP as overhead byte.
GCC141, selects GCC1 as overhead byte.
GCC142, selects GCC1 as overhead byte.
GCC243, selects GCC2 as overhead byte.
GCC244, selects GCC2 as overhead byte.
APSPCC45, selects APSPCC as overhead byte.
APSPCC46, selects APSPCC as overhead byte.
APSPCC47, selects APSPCC as overhead byte.

**:SENSe[1..n]:DATA:TELeom:OTN:OH:
ODU[1..n]:E:OVERhead?**

APSPCC48, selects APSPCC as overhead byte.

RES49, selects RES as overhead byte.

RES410, selects RES as overhead byte.

RES411, selects RES as overhead byte.

RES412, selects RES as overhead byte.

RES413, selects RES as overhead byte.

RES414, selects RES as overhead byte.

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.

Response Syntax <Value>

Response(s) Value:
The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.
Returns the overhead byte values in hexadecimal format.

Example(s) * SENS:DATA:TEL:OTN:OH:ODU1:E:OVER? RES21
Returns the ODU overhead byte values.

See Also * SOURce[1..n]:DATA:TELeom:OTN:OH:
ODU[1..n]:E:OVERhead

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:E:OVERhead****Description**

This command sets the overhead byte values for non standard rates OPU1e/2e.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:  
OPU[1..n]:E:OVERhead<wsp>RES115|RES116|  
JC116|RES215|RES216|JC216|RES315|RES316|J  
C316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|PSI5  
|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|PSI13  
|PSI14|PSI15|PSI16|PSI17,<Value>
```

:SOURce[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 <CHARACTER PROGRAM DATA> elements for this parameter are:

RES115|RES116|JC116|RES215|RES216|JC216|
RES315|RES316|JC316|PSI415|NJO416|PSI0|PS
I2|PSI3|PSI4|PSI5|PSI6|PSI7|PSI8|PSI9|PSI10|
PSI11|PSI12|PSI13|PSI14|PSI15|PSI16|PSI17.

Selects the overhead byte values for non standard rates OPU1e/2e.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:E:OVERhead

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Value:

The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the overhead byte values.

Example(s)

* SOUR:DATA:TEL:OTN:OH:OPU1:E:OVER
RES115, #HF6

* SOUR:DATA:TEL:OTN:OH:OPU1:E:OVER?
RES115 Returns #HF6

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:E:OVERhead**

See Also

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

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:E:OVERhead?****Description**

This query returns the overhead byte values for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:E:OVERhead?<wsp>RES115|RES116
|JC116|RES215|RES216|JC216|RES315|RES316
|JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|P
SI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|PS
I13|PSI14|PSI15|PSI16|PSI17

**:SOURce[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 <CHARACTER PROGRAM DATA> elements for this parameter are:

RES115|RES116|JC116|RES215|RES216|JC216|RES315|RES316|JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|PSI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|PSI13|PSI14|PSI15|PSI16|PSI17.

Selects overhead bytes.

Selects the overhead byte values for non standard rates OPU1e/2e.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

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

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Response Syntax <Value>

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

Response(s)

Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the overhead byte values in hexadecimal format.

Returns the received overhead byte values for non standard rates OPU1e/2e.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte. PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

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

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Example(s)

* SOUR:DATA:TEL:OTN:OH:OPU1:E:OVER
RES115, #HF6

* SOUR:DATA:TEL:OTN:OH:OPU1:E:OVER?
RES115 Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:E:OVERhead

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:E:DEFault**

Description	<p>This command resets or overwrites the overhead byte values for non standard rates OTU1e/2e.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:E:DEFault</p>
Parameter(s)	<p>None</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OH:OPU1:E:DEF</p>

:SENSe[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:E:OVERhead?

Description

This query returns the regenerator overhead byte values for non standard rates OTU1e/2e.

At *RST, this value is device dependent.

Syntax

```
:SENSe[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:E:OVERhead?<wsp>RES115|RES116
|JC116|RES215|RES216|JC216|RES315|RES316
|JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|P
SI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|PS
I13|PSI14|PSI15|PSI16|PSI17
```

**: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 <CHARACTER PROGRAM DATA> elements for this parameter are:

RES115|RES116|JC116|RES215|RES216|JC216|RES315|RES316|JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|PSI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|PSI13|PSI14|PSI15|PSI16|PSI17.

Selects the overhead byte values for non standard rates OPU1e/2e.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

**:SENSe[1..n]:DATA:TELecom:OTN:OH:
OPU[1..n]:E:OVERhead?**

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Response Syntax

<Value>

**:SENSe[1..n]:DATA:TELecom:OTN:OH:
OPU[1..n]:E:OVERhead?**

Response(s)	Value:
	The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.
	Returns the received overhead byte values for non standard rates OPU1e/2e.
	RES115, selects RES as overhead byte.
	RES116, selects RES as overhead byte.
	JC116, selects JC as overhead byte.
	RES215, selects RES as overhead byte.
	RES216, selects RES as overhead byte.
	JC216, selects JC as overhead byte.
	RES315, selects RES as overhead byte.
	RES316, selects RES as overhead byte.
	JC316, selects JC as overhead byte.
	PSI415, selects PSI as overhead byte.
	NJO416, selects NJO as overhead byte.
	PSI0, selects PSI0 as overhead byte.
	PSI2, selects PSI2 as overhead byte.
	PSI3, selects PSI3 as overhead byte.
	PSI4, selects PSI4 as overhead byte.
	PSI5, selects PSI5 as overhead byte.
	PSI6, selects PSI6 as overhead byte.

**:SENSe[1..n]:DATA:TELeom:OTN:OH:
OPU[1..n]:E:OVERhead?**

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Example(s)

* SENS:DATA:TEL:OTN:OH:OPU1:E:OVER?
RES115 Returns the OPU overhead byte values.

See Also

* SOURce[1..n]:DATA:TELeom:OTN:OH:
OPU[1..n]:E:OVERhead

DSn Command Reference

:SOURce[1..n]:DATA:TELEcom:DS[1..n]: PAYLoad:FRAMing

Description This command selects the framing that will be used for transmission of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3).

At *RST, this value is set to CBITp.

Syntax :SOURce[1..n]:DATA:TELEcom:DS[1..n]:
PAYLoad:FRAMing<wsp>UNFRAMED1|M13|
CBITp|SF1|ESF

Parameter(s) Framing:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
UNFRAMED1|M13|CBITp|SF1|ESF.
Sets the framing that will be used for transmission.
UNFRAMED1, selects the Unframed as DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) framing.
M13, selects the M13 as DS3 framing.
CBITp, selects the Control-Bit (CBIT) as DS3 framing.

**:SOURce[1..n]:DATA:TELEcom:DS[1..n]:
PAYLoad:FRAMing**

SF1, selects the SF (Superframe) as DS1 framing.
ESF, selects the ESF (Extended Superframe) as DS1 framing.

Example(s)

* SOUR:DATA:TEL:DS1:PAYL:FRAM UNF1
* SOUR:DATA:TEL:DS1:PAYL:FRAM?
Returns UNFRAMED1

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:DS1[1..n]:
PAYLoad:FRAMing?

:SOURCE[1..n]:DATA:TELEcom:DS[1..n]: PAYLoad:FRAMing?

Description	<p>This query returns the framing that will be used for transmission of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3).</p> <p>At *RST, this value is set to CBITp.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:DS[1..n]: PAYLoad:FRAMing?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Framing></p>
Response(s)	<p>Framing: The response data syntax for <Framing> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the framing that will be used for transmission.</p> <p>UNFRAMED1, Unframed is selected as Digital Signal-level 1 (DS1)/Digital Signal-level 3 (DS3) framing.</p> <p>M13, M13 is selected as Digital Signal-level 3 (DS3) framing.</p> <p>CBITp, Control-Bit (C-Bit) Parity is selected as Digital Signal-level 3 (DS3) framing.</p>

**:SOURce[1..n]:DATA:TELEcom:DS[1..n]:
PAYLoad:FRAMing?**

SF1, Superframe (FSF) is selected as Digital Signal-level 1 (DS1) framing.

ESF, Extended Superframe (FESF) is selected as Digital Signal-level 1 (DS1) framing.

Example(s)

* SOUR:DATA:TEL:DS1:PAYL:FRAM UNF1
* SOUR:DATA:TEL:DS1:PAYL:FRAM?
Returns UNFRAMED1

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:DS[1..n]:
PAYLoad:FRAMing

**:SENSe[1..n]:DATA:TELEcom:DS[1..n]:
PAYLoad:FRAMing**

Description	<p>This command selects the framing that will be used for received signal of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3).</p> <p>At *RST, this value is set to CBITp.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:DS[1..n]:PAYLoad:FRAMing<wsp>UNFramed1 M13 CBITp SF1 ESF</p>
Parameter(s)	<p>Framing: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: UNFramed1 M13 CBITp SF1 ESF. Sets the framing that will be used for received signal. UNFramed1, selects the Unframed as DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) framing. M13, selects the M13 as DS3 framing. CBITp, selects the Control-Bit (CBIT) as DS3 framing. SF1, selects the SF (Superframe) as DS1 framing.</p>

**:SENSe[1..n]:DATA:TELEcom:DS[1..n]:
PAYLoad:FRAMing**

ESF, selects the ESF (Extended Superframe) as DS1 framing.

Example(s)

* SENS:DATA:TEL:DS1:PAYL:FRAM UNF1
* SENS:DATA:TEL:DS1:PAYL:FRAM?
Returns UNFRAMED1

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SENSe[1..n]:DATA:TELEcom:DS[1..n]:PAYLoad:FRAMing?

**:SENSe[1..n]:DATA:TELEcom:DS[1..n]:
PAYLoad:FRAMing?**

Description	<p>This query returns the framing that will be used for received signal of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3).</p> <p>At *RST, this value is set to CBITp.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:DS[1..n]:PAYLoad:FRAMing?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Framing></p>
Response(s)	<p>Framing: The response data syntax for <Framing> is defined as a <CHARACTER RESPONSE DATA> element. Returns the framing that will be used for received signal. UNFRAMED1, Unframed is selected as Digital Signal-level 1 (DS1)/Digital Signal-level 3 (DS3) framing. M13, M13 is selected as Digital Signal-level 3 (DS3) framing. CBITp, Control-Bit (C-Bit) Parity is selected as Digital Signal-level 3 (DS3) framing.</p>

**:SENSe[1..n]:DATA:TELEcom:DS[1..n]:
PAYLoad:FRAMing?**

SF1, Superframe (FSF) is selected as Digital Signal-level 1 (DS1) framing.
ESF, Extended Superframe (FESF) is selected as Digital Signal-level 1 (DS1) framing.

Example(s)

* SENS:DATA:TEL:DS1:PAYL:FRAM UNF1
* SENS:DATA:TEL:DS1:PAYL:FRAM?
Returns UNFRAMED1

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SENSe[1..n]:DATA:TELEcom:DS[1..n]:PAYLoad:FRAMing

:SOURce[1..n]:DATA:TELEcom:DS:ENABled

Description This command enables or disables the activation of Digital Signal-level 0 (DS0)/64K testing.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:DS:ENABled
<wsp> <Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the activation of Digital Signal-level 0 (DS0)/64K testing.

Example(s) * SOUR:DATA:TEL:DS:ENAB ON
* SOUR:DATA:TEL:DS:ENAB? Returns 1

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also * SOURce[1..n]:DATA:TELEcom:DS:ENABled?

:SOURce[1..n]:DATA:TELEcom:DS:ENABLEd?

Description	This query returns the status of Digital Signal-level 0 (DS0)/64K testing. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:DS:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Digital Signal-level 0 (DS0) /64K testing.
Example(s)	* SOUR:DATA:TEL:DS:ENAB ON * SOUR:DATA:TEL:DS:ENAB? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:DS:ENABLEd

:SENSe[1..n]:DATA:TELEcom:DS:ENABLEd

Description This command enables or disables the activation of Digital Signal-level 0 (DS0)/64K testing.

At *RST, this value is set to OFF.

Syntax :SENSe[1..n]:DATA:TELEcom:DS:ENABLEd
<wsp> <Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the activation of Digital Signal-level 0 (DS0)/64K testing.

Example(s) * SENS:DATA:TEL:DS:ENAB ON
* SENS:DATA:TEL:DS:ENAB? Returns 1

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also * SENSe[1..n]:DATA:TELEcom:DS:ENABLEd?

:SENSe[1..n]:DATA:TELEcom:DS:ENABled?

Description	This query returns the status of Digital Signal-level 0 (DS0)/64K testing. At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:DS:ENABled?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Digital Signal-level 0 (DS0)/64K testing.
Example(s)	* SENS:DATA:TEL:DS:ENAB ON * SENS:DATA:TEL:DS:ENAB? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SENSe[1..n]:DATA:TELEcom:DS:ENABled

:SOURce[1..n]:DATA:TELEcom:DS:MODE

Description	<p>This command sets the channel timeslot data rate of the pattern payload content for transmitter.</p> <p>At *RST, this value is set to E64K.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DS:MODE<wsp> E64K E56K</pre>
Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: E64K E56K.</p> <p>Sets the channel timeslot data rate for the pattern payload content.</p> <p>E64K, selects a timeslot data rate of E64 Kbps uses 8 bits to carry the payload information.</p> <p>E56K, selects a timeslot data rate of E56 Kbps uses 7 bits to carry the payload information.</p>

:SOURce[1..n]:DATA:TELEcom:DS:MODE**Example(s)**

- * SOUR:DATA:TEL:DS:ENAB ON
- * SOUR:DATA:TEL:DS:MOD E64K
- * SOUR:DATA:TEL:DS:MOD? Returns E64K

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:DS:ENABled
 - * SOURce[1..n]:DATA:TELEcom:DS:MODE?
-

:SOURce[1..n]:DATA:TELEcom:DS:MODE?

Description This query returns the channel timeslot data rate of the pattern payload content for transmitter.

At *RST, this value is set to E64K.

Syntax :SOURce[1..n]:DATA:TELEcom:DS:MODE?

Parameter(s) None

Response Syntax <Mode>

Response(s) Mode:
The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the channel timeslot data rate for the pattern payload content.
E64K, timeslot data rate of 64 Kbps uses 8 bits to carry the payload information is selected.
E56K, timeslot data rate of 56 Kbps uses 7 bits to carry the payload information is selected.

:SOURce[1..n]:DATA:TELEcom:DS:MODE?**Example(s)**

- * SOUR:DATA:TEL:DS:ENAB ON
- * SOUR:DATA:TEL:DS:MOD E64K
- * SOUR:DATA:TEL:DS:MOD? Returns E64K

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:DS:ENABled
 - * SOURce[1..n]:DATA:TELEcom:DS:MODE
-

:SENSe[1..n]:DATA:TELEcom:DS:MODE

Description	<p>This command sets the channel timeslot data rate for receiver.</p> <p>At *RST, this value is set to E64K.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:DS:MODE<wsp> E64K E56K</p>
Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: E64K E56K.</p> <p>Sets the channel timeslot data rate for the pattern payload content.</p> <p>E64K, selects the timeslot data rate of E64 Kbps uses 8 bits to carry the payload information.</p> <p>E56K, selects the timeslot data rate of E56 Kbps uses 7 bits to carry the payload information.</p>

:SENSe[1..n]:DATA:TELecom:DS:MODE**Example(s)**

- * SENS:DATA:TEL:DS:ENAB ON
- * SENS:DATA:TEL:DS:MODE E64K
- * SENS:DATA:TEL:DS:MODE? Returns E64K

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SENSe[1..n]:DATA:TELecom:DS:ENABLEd
 - * SENSe[1..n]:DATA:TELecom:DS:MODE?
-

:SENSe[1..n]:DATA:TELEcom:DS:MODE?

Description This query returns the channel timeslot data rate for receiver.

At *RST, this value is set to E64K.

Syntax :SENSe[1..n]:DATA:TELEcom:DS:MODE?

Parameter(s) None

Response Syntax <Mode>

Response(s) Mode:
The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the channel timeslot data rate for the pattern payload content.
E64K, timeslot data rate of 64 Kbps uses 8 bits to carry the payload information is selected.
E56K, timeslot data rate of 56 Kbps uses 7 bits to carry the payload information is selected.

:SENSe[1..n]:DATA:TELecom:DS:MODE?**Example(s)**

- * SENS:DATA:TEL:DS:ENAB ON
- * SENS:DATA:TEL:DS:MODE E64K
- * SENS:DATA:TEL:DS:MODE? Returns E64K

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SENSe[1..n]:DATA:TELecom:DS:ENABled
 - * SENSe[1..n]:DATA:TELecom:DS:MODE
-

:SOURCE[1..n]:DATA:TELEcom:DS:ZCS

Description This command sets the Zero Code Suppression (ZCS) method used to replace the all-zero bytes of the idle and tone payload contents.

At *RST, this value is set to NONE.

Syntax :SOURCE[1..n]:DATA:TELEcom:DS:ZCS<wsp>
NONE|JBIT8|GTE|ZBELI

Parameter(s) Zcs:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
NONE|JBIT8|GTE|ZBELI.
Sets the Zero Code Suppression (ZCS) method used to replace the all-zero bytes of the idle and tone payload contents.
NONE, No Zero Code Suppression (ZCS) method is selected.
JBIT8, selects every 8th (LSB) bit is forced to 1.
GTE, selects bit 8 of an all zero channel byte is replaced by 1, except in signaling frames where bit 7 is forced to 1.
ZBELI, selects bit 7 of an all zero channel byte is replaced by 1.

:SOURce[1..n]:DATA:TELEcom:DS:ZCS**Example(s)**

- * SOUR:DATA:TEL:DS:ENAB ON
- * SOUR:DATA:TEL:DS:ZCS NONE
- * SOUR:DATA:TEL:DS:ZCS? Returns NONE

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:DS:ENABled
 - * SOURce[1..n]:DATA:TELEcom:DS:ZCS?
-

:SOURce[1..n]:DATA:TELEcom:DS:ZCS?

Description This query returns the Zero Code Suppression (ZCS) method used to replace the all-zero bytes of the idle and tone payload contents.

At *RST, this value is set to NONE.

Syntax :SOURce[1..n]:DATA:TELEcom:DS:ZCS?

Parameter(s) None

Response Syntax <Zcs>

Response(s) Zcs:
The response data syntax for <Zcs> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Zero Code Suppression (ZCS) method used to replace the all-zero bytes of the idle and tone payload contents.
NONE, No Zero Code Suppression (ZCS) method is selected.
JBIT8, every 8th (LSB) bit is forced to 1 is selected.
GTE, bit 8 of an all zero channel byte is replaced by 1, except in signaling frames where bit 7 is forced to 1 is selected.
ZBELL, bit 7 of an all zero channel byte is replaced by 1 is selected.

:SOURce[1..n]:DATA:TELEcom:DS:ZCS?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DS:ENAB ON* SOUR:DATA:TEL:DS:ZCS NONE* SOUR:DATA:TEL:DS:ZCS? Returns NONE
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DS:ENABled* SOURce[1..n]:DATA:TELEcom:DS:ZCS

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
IDLE**

Description	<p>This command sets the idle code byte from the idle field. The selected idle code applies to all timeslots set to idle.</p> <p>At *RST, this value is set to #H7F.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: IDLE <wsp> <Idle></p>
Parameter(s)	<p>Idle:</p> <p>The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the idle code byte from the idle field.</p> <p>The values are hexadecimal #H00 to #HFF.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DS:ENAB ON* SOUR:DATA:TEL:DS:PAYL:IDLE #H70* SOUR:DATA:TEL:DS:PAYL:IDLE? Returns #H70
Note	<p>FTB/IQS-8140 Transport Blazer does not support this command.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DS:ENABLEd* SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:IDLE?

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
IDLE?**

Description	<p>This query returns the idle code byte from the idle field. The selected idle code applies to all timeslots set to Idle.</p> <p>At *RST, this value is set to #H7F.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: IDLE?
Parameter(s)	None
Response Syntax	<Idle>
Response(s)	<p>Idle:</p> <p>The response data syntax for <Idle> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the idle code byte from the idle field. The values are hexadecimal #H00 to #HFF.</p>

:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: IDLE?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DS:ENAB ON* SOUR:DATA:TEL:DS:PAYL:IDLE #H70* SOUR:DATA:TEL:DS:PAYL:IDLE? Returns #H70
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DS:ENABled* SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:IDLE

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
TONE**

Description	<p>This command sets the tone for digital milli watt testing. The signal output power, when converted to analog, is 0 dBm. The selected tone applies to all timeslots set to tone.</p> <p>At *RST, this value is set to T1004.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: TONE<wsp>T1000 T1004</pre>
Parameter(s)	<p>Tone:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: T1000 T1004.</p> <p>Sets the tone for digital milli watt testing.</p> <p>T1000, selects the 1000 Hz tone.</p> <p>T1004, selects the 1004 Hz tone.</p>

:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: TONE

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DS:ENAB ON* SOUR:DATA:TEL:DS:PAYL:TONE T1000* SOUR:DATA:TEL:DS:PAYL:TONE? Returns T1000
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DS:ENABled* SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:TONE?

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
TONE?**

Description	<p>This query returns the tone for digital milli watt testing.</p> <p>At *RST, this value is set to T1004.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: TONE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Tone></p>
Response(s)	<p>Tone: The response data syntax for <Tone> is defined as a <CHARACTER RESPONSE DATA> element. Returns the tone for digital milli watt testing. T1000, 1000 Hz is selected as tone. T1004, 1004 Hz is selected as tone.</p>

:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: TONE?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DS:ENAB ON* SOUR:DATA:TEL:DS:PAYL:TONE T1000* SOUR:DATA:TEL:DS:PAYL:TONE? Returns T1000
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DS:ENABled* SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:TONE

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent**

Description	<p>This command sets the payload content for the transmitter.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent<wsp><Content> ,PATTERN1 IDLE TONE</pre>
Parameter(s)	<p>Content: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the timeslots for the payload content. Choices are 1 through 24.</p> <p>Payload: The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PATTERN1 IDLE TONE. Sets the payload content. PATTERN1, selects the Pattern as payload content. IDLE, selects the Idle as payload content. TONE, selects the Tone as payload content.</p>

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DS:ENAB ON* SOUR:DATA:TEL:DS:PAYL:CONT 10,PAT1* SOUR:DATA:TEL:DS:PAYL:CONT? 10 Returns PATTERN1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DS:ENABled* SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent?

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent?**

Description	<p>This query returns the payload content for the transmitter.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent? <wsp> <Content></p>
Parameter(s)	<p>Content: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the timeslots for the payload content. Choices are 1 through 24.</p>
Response Syntax	<p><Payload></p>
Response(s)	<p>Payload: The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element. Returns the payload content. PATTERN1, Pattern is selected as payload content. IDLE, Idle is selected as payload content. TONE, Tone is selected as payload content.</p>

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DS:ENAB ON* SOUR:DATA:TEL:DS:PAYL:CONT 10,PAT1* SOUR:DATA:TEL:DS:PAYL:CONT? 10 Returns PATTERN1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DS:ENABled* SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent

**:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent**

Description	<p>This command sets the payload content for the receiver.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent<wsp> <Content> ,NONE PATTERN1</pre>
Parameter(s)	<p>Content: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the timeslots for payload content. Choices are 1 through 24.</p> <p>Payload: The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NONE PATTERN1. Sets the payload content. NONE, No payload content is selected. PATTERN1, selects the Pattern as payload content.</p>

:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:DS:ENAB ON* SENS:DATA:TEL:DS:PAYL:CONT 10,NONE* SENS:DATA:TEL:DS:PAYL:CONT? 10 Returns NONE
Note	<ul style="list-style-type: none">* Payload content configuration is only available for decoupled test mode.* FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SENSE[1..n]:DATA:TELEcom:DS:ENABled* SENSE[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent

**:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent?**

Description	<p>This query returns the payload content for the receiver.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent? <wsp> <Content></p>
Parameter(s)	<p>Content: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the timeslots for payload content. Choices are 1 through 24.</p>
Response Syntax	<p><Payload></p>
Response(s)	<p>Payload: The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element. Returns the payload content. NONE, No payload content is selected. PATTERN1, Pattern is selected as payload content.</p>

:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent?

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:DS:ENAB ON* SENS:DATA:TEL:DS:PAYL:CONT 10,NONE* SENS:DATA:TEL:DS:PAYL:CONT? 10 Returns NONE
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:DS:ENABled* SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent:TYPE**

Description	<p>This command selects type of payload content for transmitter.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:TYPE<wsp>PATTERN1 IDLE TONE</pre>
Parameter(s)	<p>Payload:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PATTERN1 IDLE TONE.</p> <p>Sets the payload content type.</p> <p>PATTERN1, selects the Pattern as payload type.</p> <p>IDLE, selects the Idle as payload type.</p> <p>TONE, selects the Tone as payload type.</p>

:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:TYPE

Example(s)

- * SOUR:DATA:TEL:DS:ENAB ON
- * SOUR:DATA:TEL:DS:PAYL:CONT:TYPE PAT1
- * SOUR:DATA:TEL:DS:PAYL:CONT:TYPE? Returns PATTERN1

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:DS:ENABled
- * SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent:TYPE?
- * SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent:ALL

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent:TYPE?**

Description	<p>This query returns the type of payload content for the transmitter.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:TYPE?
Parameter(s)	None
Response Syntax	<Payload>
Response(s)	<p>Payload:</p> <p>The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the payload content type.</p> <p>PATTERN1, Pattern is selected as payload type.</p> <p>IDLE, Idle is selected as payload type.</p> <p>TONE, Tone is selected as payload type.</p>

:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:TYPE?

Example(s)

- * SOUR:DATA:TEL:DS:ENAB ON
- * SOUR:DATA:TEL:DS:PAYL:CONT:TYPE PAT1
- * SOUR:DATA:TEL:DS:PAYL:CONT:TYPE? Returns PATTERN1

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:DS:ENABled
- * SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent:TYPE
- * SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent:ALL

**:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent:TYPE****Description**

This command selects the type of payload content for receiver.

At *RST, this value is set to PATTERN1.

Syntax

:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent:TYPE<wsp>NONE|PATTERN1

Parameter(s)

Payload:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

NONE|PATTERN1.

Sets the type of payload content.

NONE, No type of payload content is selected.

PATTERN1, selects Pattern as type of payload content.

:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:TYPE

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:DS:ENAB ON* SENS:DATA:TEL:DS:PAYL:CONT:TYPE NONE* SENS:DATA:TEL:DS:PAYL:CONT:TYPE? Returns NONE
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:DS:ENABled* SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent:TYPE?* SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent:ALL

**:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent:TYPE?**

Description	<p>This query returns type of payload content for the receiver.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:TYPE?
Parameter(s)	None
Response Syntax	<Payload>
Response(s)	<p>Payload:</p> <p>The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of payload content.</p> <p>NONE, No type of payload content is selected.</p> <p>PATTERN1, Pattern is selected as type of payload content.</p>

:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:TYPE?

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:DS:ENAB ON* SENS:DATA:TEL:DS:PAYL:CONT:TYPE NONE* SENS:DATA:TEL:DS:PAYL:CONT:TYPE? Returns NONE
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:DS:ENABled* SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent:TYPE* SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent:ALL

**:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent:ALL**

Description	<p>This command sets the payload content of all timeslots to the selected payload content with its Pattern, Idle, or Tone value.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:ALL
Parameter(s)	None

:SOURce[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:ALL

Example(s)

- * SOUR:DATA:TEL:DS:ENAB ON
- * SOUR:DATA:TEL:DS:PAYL:CONT:TYPE PAT1
- * SOUR:DATA:TEL:DS:PAYL:CONT:ALL

Note

- * The timeslots set to Idle or Tone can be changed from Idle to Tone and vice versa even when the test is running. The Idle and Tone values can also be changed.
- * FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:DS:ENABled
- * SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent:TYPE
- * SOURce[1..n]:DATA:TELEcom:DS:PAYLoad:CONTent:TYPE?

**:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad:
CONTent:ALL**

Description	<p>This command sets the payload content of all timeslots with (Pattern) or without (None) the selected pattern.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:ALL
Parameter(s)	None
Example(s)	* SENS:DATA:TEL:DS:ENAB ON * SENS:DATA:TEL:DS:PAYL:CONT:TYPE NONE * SENS:DATA:TEL:DS:PAYL:CONT:ALL
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SENSe[1..n]:DATA:TELEcom:DS:ENABLEd * SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:TYPE * SENSe[1..n]:DATA:TELEcom:DS:PAYLoad: CONTent:TYPE?

**:SOURce[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:TYPE**

Description This command selects the type of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) alarm.

At *RST, this value is set to AIS.

Syntax :SOURce[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:TYPE <wsp>AIS|RAI|OOF|RDI1|IDLE1

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
AIS|RAI|OOF|RDI1|IDLE1.
Selects the type of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3) alarm.
AIS, selects AIS (Alarm Indication Signal) alarm when an unframed all-ones signal is received for DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3).
RAI, selects RAI (Remote Alarm Indicates) for DS1 (Digital Signal-level 1) which shows the following:

**:SOURce[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:TYPE**

For SF framing: The RAI alarm is declared when bit 2 in each timeslot contains "0".

For ESF framing: The RAI alarm is declared when eight "ones" followed by eight "zeros" pattern is received continuously in the data link (FDL).

OOF, selects OOF (Out of Frame) error which indicates that four consecutive frame bit errors are detected for DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3).

RDI1, selects RDI (Remote Defect Indicator) alarm is declared when both X-bits of the M-Frame are set to "0" for (Digital Signal-level 3).

IDLE1, Idle alarm is declared when subframe 3 of the M-frame contains zeros (0) for the three C-bits, ones (1) for X-bits, 1100... repeating sequence with the first two bits following each control bit set to 11 for the information bits.

Example(s)

* SOUR:DATA:TEL:DSN:ALAR:DS1:TYPE AIS

* SOUR:DATA:TEL:DSN:ALAR:DS1:TYPE?

Returns AIS

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:TYPE?

**:SOURce[1..n]:DATA:TELEcom:DSN:ALARM:
DS[1..n]:TYPE?**

Description	<p>This query returns the type of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) alarm.</p> <p>At *RST, this value is set to AIS.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:DSN:ALARM: DS[1..n]:TYPE?</code>
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Digital Signal-level 1 (DS1) /Digital Signal-level 3 (DS3) alarm.</p> <p>AIS, Alarm Indication Signal (AIS) is selected as Digital Signal-level 1(DS1)/Digital Signal-level 3 (DS3) alarm.</p> <p>RAI, Remote Alarm Indicates (RAI) is selected as Digital Signal-level 1(DS1) alarm.</p> <p>OOF, Out of Frame (OOF) is selected as Digital Signal-level 1(DS1)/Digital Signal-level 3 (DS3) alarm.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:TYPE?**

RD11, Remote Defect Indicator (RDI) is selected as Digital Signal-level 3 (DS3) alarm.
IDLE1, Idle is selected as Digital Signal-level 3 (DS3) alarm.

Example(s)

* SOUR:DATA:TEL:DSN:ALAR:DS1:TYPE AIS
* SOUR:DATA:TEL:DSN:ALAR:DS1:TYPE?
Returns AIS

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:TYPE

:SOURce[1..n]:DATA:TELEcom:DSN:ALARM:DS[1..n]

Description	<p>This command enables or disables the DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DSN:ALARM:DS[1..n]<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) alarm generation.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:ALAR:DS1:TYPE AIS* SOUR:DATA:TEL:DSN:ALAR:DS1 ON* SOUR:DATA:TEL:DSN:ALAR:DS1? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:ALARm:DS[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:DSN:ALARm:DS[1..n]?

:SOURce[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]?

Description	<p>This query returns the status of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) alarm generation.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Digital Signal-level 1 (DS1) /Digital Signal-level 3 (DS3) alarm generation.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]?****Example(s)**

- * SOUR:DATA:TEL:DSN:ALAR:DS1:TYPE AIS
- * SOUR:DATA:TEL:DSN:ALAR:DS1 ON
- * SOUR:DATA:TEL:DSN:ALAR:DS1? Returns 1

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:DSN:ALARm:DS[1..n]:TYPE
 - * SOURce[1..n]:DATA:TELEcom:DSN:ALARm:DS[1..n]
-

:SOURce[1..n]:DATA:TELEcom:DSN:ERROr: DS[1..n]:MANual:TYPE

Description This command selects the manual type of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:DSN:ERROr:
DS[1..n]:MANual:TYPE<wsp>CRC6|FBIT|CBIT|
FBIT1|PBIT|FEBE

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
CRC6|FBIT|CBIT|FBIT1|PBIT|FEBE.
Selects the type of DS1(Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.
CRC6, selects the CRC-6 (Cyclical Redundancy Check-6) error which indicates that one or more bit errors have been detected in a block of data through cyclical redundancy check for DS1 (Digital Signal-level 1).

**:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR:
DS[1..n]:MANual:TYPE**

FBIT, selects the F-Bit (Framing Bit) error which indicates that an incorrect value appeared in a bit position reserved for DS1 (Digital Signal-level 1) framing.

CBIT, selects the CBIT (Control-Bit) error which indicates that the three C-bits reserved to control bit stuffing are different of "111" and "000" for DS3 (Digital Signal-level 3).

FBIT1, selects the DS3 (Digital Signal-level 3) Framing-Bit (FBIT1) error which indicates that the frame alignment pattern received is different for "1001".

PBIT, selects the Parity-Bit (PBIT) error which indicates that the P-Bits does not match the parity of all the information bits following the first X-Bit of the previous DS3 (Digital Signal-level 3) frame.

FEBE, selects the Far-End Block Error (FEBE) when the three FEBE bits reserved for DS3 (Digital Signal-level 3) framing or parity error detection contain the "000" pattern.

**:SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:MANual:TYPE**

Example(s) * SOUR:DATA:TEL:DSN:ERR:DS1:MAN:TYPE
CRC6
* SOUR:DATA:TEL:DSN:ERR:DS1:MAN:TYPE?
Returns CRC6

Note FTB/IQS-8140 Transport Blazer does not support
this command.

See Also * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:MANual:TYPE?

**:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR:
DS[1..n]:MANual:TYPE?**

Description	<p>This query returns the manual type of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR: DS[1..n]:MANual:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Digital Signal-level 1(DS1) /Digital Signal-level 3 (DS3) error. CRC6, Cyclic Redundancy Check is selected as Digital Signal-level 1 (DS1) error. FBIT, Framing Bit (FBIT) is selected as Digital Signal-level 1(DS1) error.</p>

:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:MANual:TYPE?

CBIT, Control-Bit (C-BIT) is selected as Digital Signal-level 3 (DS3) error.

FBIT1, Framing bit (FBIT) is selected as Digital Signal-level 3 (DS3) error.

PBIT, Parity bit (P-BIT) is selected as Digital Signal-level 3 (DS3) error.

FEBE, Far-End Block Error (FEBE) is selected as Digital Signal-level 3 (DS3) error.

Example(s)

* SOUR:DATA:TEL:DSN:ERR:DS1:MAN:TYPE
CRC6

* SOUR:DATA:TEL:DSN:ERR:DS1:MAN:TYPE?
Returns CRC6

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:AMOut**

Description	<p>This command sets the amount of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error to be injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AMOut<wsp> <Amount> MAXimum MINimum</pre>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the amount of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3) error.</p> <p>Choices are 1 through 50.</p>

:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AMOunt

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:ERR:DS1:MAN:TYPE CRC6* SOUR:DATA:TEL:DSN:ERR:DS1:AMO 15* SOUR:DATA:TEL:DSN:ERR:DS1:AMO? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AMOunt?

**:SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:AMOUnt?**

Description	<p>This query returns the amount of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AMOUnt? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<pre><Amount></pre>

:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AMOut?

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Digital Signal-level 1 (DS1) /Digital Signal-level 3 (DS3) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:DSN:ERR:DS1:MAN:TYPE CRC6</p> <p>* SOUR:DATA:TEL:DSN:ERR:DS1:AMO 15</p> <p>* SOUR:DATA:TEL:DSN:ERR:DS1:AMO?</p> <p>Returns 15</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AMOut</p>

**:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR:
DS[1..n]:INJECT**

Description	<p>This command injects the type of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error into the instrument.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR: DS[1..n]:INJECT
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:DSN:ERR:DS1:MAN:TYPE CRC6 * SOUR:DATA:TEL:DSN:ERR:DS1:AMO 15 * SOUR:DATA:TEL:DSN:ERR:DS1:INJ
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURCE[1..n]:DATA:TELEcom:DSN:ERROR:DS1: MANual:TYPE * SOURCE[1..n]:DATA:TELEcom:DSN:ERROR:DS1: AMOUNT

**:FETCh[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:HISTory?**

Description	<p>This query returns the history status of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:HISTory?<wsp>CRC6 FBIT CBIT FBIT1 PBIT FEBE</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CRC6 FBIT CBIT FBIT1 PBIT FEBE.</p> <p>Selects the type of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.</p> <p>CRC6, selects the CRC-6 (Cyclical Redundancy Check-6) error which indicates that one or more bit errors have been detected in a block of data through cyclical redundancy check for DS1 (Digital Signal-level 1).</p> <p>FBIT, selects the F-Bit (Framing Bit) error which indicates that an incorrect value appeared in a bit position reserved for DS1 (Digital Signal-level 1) framing.</p>

:FETCh[1..n]:DATA:TELEcom:DSN:ERROr: DS[1..n]:HISTOrY?

CBIT, selects the Control-Bit (CBIT) error which indicates that the three C-bits reserved to control bit stuffing are different of "111" and "000" for DS3 (Digital Signal-level 3).

FBIT1, selects the DS3 Framing-Bit (FBIT1) error which indicates that the frame alignment pattern received is different for "1001".

PBIT, selects the Parity-Bit (PBIT) error which indicates that the P-Bits does not match the parity of all the information bits following the first X-Bit of the previous DS3 frame.

FEBE, selects the Far-End Block Error (FEBE) when the three FEBE bits reserved for DS3 (Digital Signal-level 3) framing or parity error detection contain the "000" pattern.

Response Syntax <History>

Response(s) History:
The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the history status Digital Signal-level 1 (DS1)/Digital Signal-level 3 (DS3) 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.

:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:HISTory?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:ERR:DS1:MAN:TYPE CRC6* SOUR:DATA:TEL:DSN:ERR:DS1:AMO 15* SOUR:DATA:TEL:DSN:ERR:DS1:INJ* FETC:DATA:TEL:DSN:ERR:DS1:HIST? CRC6 Returns the error history status.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AMOunt* SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:SEConds?**

Description	<p>This query returns the number of seconds within which DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:SEConds?<wsp>CRC6 FBIT CBIT FBIT1 PBIT FEBE</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CRC6 FBIT CBIT FBIT1 PBIT FEBE.</p> <p>Selects the type of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3) error.</p> <p>CRC6, selects the CRC-6 (Cyclical Redundancy Check-6) error which indicates that one or more bit errors have been detected in a block of data through cyclical redundancy check for DS1.</p> <p>FBIT, selects the F-Bit (Framing Bit) error which indicates that an incorrect value appeared in a bit position reserved for DS1 framing.</p>

**:FETCh[1..n]:DATA:TELEcom:DSN:ERROr:
DS[1..n]:SEConds?**

CBIT, selects the Control-Bit (CBIT) error which indicates that the three C-bits reserved to control bit stuffing are different of "111" and "000" for DS3. FBIT1, selects the DS3 Framing-Bit (FBIT1) error which indicates that the frame alignment pattern received is different for "1001".

PBIT, selects the Parity-Bit (PBIT) error which indicates that the P-Bits does not match the parity of all the information bits following the first X-Bit of the previous DS3 frame.

FEBE, selects the Far-End Block Error (FEBE) when the three FEBE bits reserved for DS3 framing or parity error detection contain the "000" pattern.

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Digital Signal-level 1 (DS1)/Digital Signal-level 3 (DS3) error.

**:FETCh[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:SEConds?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:ERR:DS1:MAN:TYPE CRC6* SOUR:DATA:TEL:DSN:ERR:DS1:AMO 15* SOUR:DATA:TEL:DSN:ERR:DS1:INJ* FETC:DATA:TEL:DSN:ERR:DS1:SEC? CRC6 Returns the number of errored seconds.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AMOUNT* SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:DSN:ERROr:
DS[1..n]:CURREnt?**

Description	<p>This query returns the current status of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:DSN:ERROr: DS[1..n]:CURREnt?<wsp>CRC6 FBIT CBIT FBIT1 PBIT FEBE</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CRC6 FBIT CBIT FBIT1 PBIT FEBE.</p> <p>Selects the type of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3) error.</p> <p>CRC6, selects the CRC-6 (Cyclical Redundancy Check-6) error which indicates that one or more bit errors have been detected in a block of data through cyclical redundancy check for DS1 (Digital Signal-level 1).</p> <p>FBIT, selects the F-Bit (Framing Bit) error which indicates that an incorrect value appeared in a bit position reserved for DS1 (Digital Signal-level 1) framing.</p>

**:FETCh[1..n]:DATA:TELEcom:DSN:ERROr:
DS[1..n]:CURRENT?**

CBIT, selects the Control-Bit (CBIT) error which indicates that the three C-bits reserved to control bit stuffing are different of "111" and "000" for DS3. FBIT1, selects the DS3 Framing-Bit (FBIT1) error which indicates that the frame alignment pattern received is different for "1001".

PBIT, selects the Parity-Bit (PBIT) error which indicates that the P-Bits does not match the parity of all the information bits following the first X-Bit of the previous DS3 (Digital Signal-level 3) frame.

FEBE, selects the Far-End Block Error (FEBE) when the three FEBE bits reserved for DS3 (Digital Signal-level 3) framing or parity error detection contain the "000" pattern.

Response Syntax <Current>

Response(s) Current:
The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the current status of 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.

:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:CURRent?

Example(s) * SOUR:DATA:TEL:DSN:ERR:DS1:MAN:TYPE
CRC6
* SOUR:DATA:TEL:DSN:ERR:DS1:AMO 15
* SOUR:DATA:TEL:DSN:ERR:DS1:INJ
* FETC:DATA:TEL:DSN:ERR:DS1:CURR? CRC6
Returns the current error status.

Note FTB/IQS-8140 Transport Blazer does not support
this query.

See Also * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:AMOunt
* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:COUNT?**

Description	<p>This query returns the count for DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:COUNT? <wsp>CRC6 FBIT CBIT FBIT1 PBIT FEBE</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CRC6 FBIT CBIT FBIT1 PBIT FEBE.</p> <p>Selects the type of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3) error.</p> <p>CRC6, selects the CRC-6 (Cyclical Redundancy Check-6) error which indicates that one or more bit errors have been detected in a block of data through cyclical redundancy check for DS1 (Digital Signal-level 1).</p> <p>FBIT, selects the F-Bit (Framing Bit) error which indicates that an incorrect value appeared in a bit position reserved for DS1 (Digital Signal-level 1) framing.</p>

**:FETCh[1..n]:DATA:TELEcom:DSN:ERROr:
DS[1..n]:COUNT?**

CBIT, selects the Control-Bit (CBIT) error which indicates that the three C-bits reserved to control bit stuffing are different of "111" and "000" for DS3.

FBIT1, selects the DS3 Framing-Bit error which indicates that the frame alignment pattern received is different for "1001".

PBIT, selects the Parity-Bit (PBIT) error which indicates that the P-Bits does not match the parity of all the information bits following the first X-Bit of the previous DS3 (Digital Signal-level 3) frame.

FEBE, selects the Far-End Block Error (FEBE) when the three FEBE bits reserved for DS3 (Digital Signal-level 3) framing or parity error detection contain the "000" pattern.

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of Digital Signal-level 1 (DS1) /Digital Signal-level 3 (DS3) error.

**:FETCh[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:COUNT?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:ERR:DS[1..n]:MAN:TYPE CRC6* SOUR:DATA:TEL:DSN:ERR:DS[1..n]:AMO 15* SOUR:DATA:TEL:DSN:ERR:DS[1..n]:INJ* FETC:DATA:TEL:DSN:ERR:DS[1..n]:COUN? CRC6 Returns the error count.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AMOunt* SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:INJect

:FETCh[1..n]:DATA:TELEcom:DSN:ERROr: DS[1..n]:RATE?

Description	<p>This query returns the current rate of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:DSN:ERROr: DS[1..n]:RATE?<wsp>CRC6 FBIT CBIT FBIT1 PBIT FEBE</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CRC6 FBIT CBIT FBIT1 PBIT FEBE.</p> <p>Selects the type of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3) error.</p> <p>CRC6, selects the CRC-6 (Cyclical Redundancy Check-6) error which indicates that one or more bit errors have been detected in a block of data through cyclical redundancy check for DS1 (Digital Signal-level 1).</p> <p>FBIT, selects the F-Bit (Framing Bit) error which indicates that an incorrect value appeared in a bit position reserved for DS1 (Digital Signal-level 1) framing.</p>

**:FETCh[1..n]:DATA:TELEcom:DSN:ERROr:
DS[1..n]:RATE?**

CBIT, selects the Control-Bit (CBIT) error which indicates that the three C-bits reserved to control bit stuffing are different of "111" and "000" for DS3. FBIT1, selects the DS3 Framing-Bit (FBIT1) error which indicates that the frame alignment pattern received is different for "1001".

PBIT, selects the Parity-Bit (PBIT) error which indicates that the P-Bits does not match the parity of all the information bits following the first X-Bit of the previous DS3 (Digital Signal-level 3) frame.

FEBE, selects the Far-End Block Error (FEBE) when the three FEBE bits reserved for DS3 (Digital Signal-level 3) framing or parity error detection contain the "000" pattern.

Response Syntax <Rate>

Response(s) Rate:
The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.
Returns the current rate of Digital Signal-level 1 (DS1)/Digital Signal-level 3 (DS3) error.

:FETCh[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:RATE?

Example(s)

- * SOUR:DATA:TEL:DSN:ERR:DS[1..n]:MAN:TYPE
CRC6
- * SOUR:DATA:TEL:DSN:ERR:DS[1..n]:AMO 15
- * SOUR:DATA:TEL:DSN:ERR:DS[1..n]:INJ
- * FETC:DATA:TEL:DSN:ERR:DS[1..n]:RATE? CRC6

Returns the error rate.

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:MANual:TYPE
- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:AMOUNT
- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:HISTory?**

Description	<p>This query returns the history status of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]:HISTory?<wsp>AIS RAI OOF RDI1 IDLE1</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>AIS RAI OOF RDI1 IDLE1.</p> <p>Selects the type of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3) alarm.</p> <p>AIS, selects Alarm Indication Signal (AIS) alarm when an unframed all-ones signal is received for DS1/DS3.</p> <p>RAI, selects Remote Alarm Indicates (RAI) DS1 (Digital Signal-level 1) which shows the following:</p> <p>For SF framing: The RAI alarm is declared when bit 2 in each timeslot contains "0".</p>

**:FETCh[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:HISTory?**

For ESF framing: The RAI alarm is declared when eight "ones" followed by eight "zeros" pattern is received continuously in the data link (FDL).

OOE, selects Out of Frame (OOE) error which indicates that four consecutive frame bit errors are detected for DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3).

RDI1, Remote Defect Indicator alarm is declared when both X-bits of the M-Frame are set to "0" for DS3 (Digital Signal-level 3).

IDLE1, Idle alarm is declared when subframe 3 of the M-frame contains zeros (0) for the three C-bits, ones (1) for X-bits, 1100... repeating sequence with the first two bits following each control bit set to 11 for the information bits.

Response Syntax <History>

Response(s) History:
The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the history status of Digital Signal-level 1 (DS1)/Digital Signal-level 3 (DS3) 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.

**:FETCh[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:HISTory?****Example(s)**

- * SOUR:DATA:TEL:DSN:ALAR:DS1:TYPE AIS
- * SOUR:DATA:TEL:DSN:ALAR:DS1 ON
- * FETC:DATA:TEL:DSN:ALAR:DS1:HIST? AIS

Returns the alarm history status.**Note**

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:DSN:ALARm:DS[1..n]:TYPE
- * SOURce[1..n]:DATA:TELEcom:DSN:ALARm:DS[1..n]

**:FETCh[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:SEConds?**

Description This query returns the number of seconds within which DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) alarm occurred.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:SEConds?<wsp>AIS|RAI|OOF|RDI1|
IDLE1

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
AIS|RAI|OOF|RDI1|IDLE1.
Selects the type of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3) alarm.
AIS, selects Alarm Indication Signal (AIS) alarm when an unframed all-ones signal is received for DS1/DS3.
RAI, selects Remote Alarm Indicates (RAI) DS1 (Digital Signal-level 1) which shows the following:
For SF framing: The RAI alarm is declared when bit 2 in each timeslot contains "0".

**:FETCh[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:SEConds?**

For ESF framing: The RAI alarm is declared when eight "ones" followed by eight "zeros" pattern is received continuously in the data link (FDL).

OOE, selects Out of Frame (OOE) error which indicates that four consecutive frame bit errors are detected for DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3).

RDI1, Remote Defect Indicator alarm is declared when both X-bits of the M-Frame are set to "0" for DS3 (Digital Signal-level 3).

IDLE1, Idle alarm is declared when subframe 3 of the M-frame contains zeros (0) for the three C-bits, ones (1) for X-bits, 1100... repeating sequence with the first two bits following each control bit set to 11 for the information bits.

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Digital Signal-level 1 (DS1)/Digital Signal-level 3 (DS3) alarm.

:FETCh[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]:SEConds?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:ALAR:DS1:TYPE AIS* SOUR:DATA:TEL:DSN:ALAR:DS1 ON* FETC:DATA:TEL:DSN:ALAR:DS1:SEC? AIS Returns the number of seconds of DS1/DS3 alarm.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:ALARm:DS[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:DSN:ALARm:DS[1..n]

**:FETCh[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:CURRent?**

Description	<p>This query returns the current status of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:DSN:ALARm: DS[1..n]:CURRent?<wsp>AIS RAI OOF RDI1 IDLE1</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>AIS RAI OOF RDI1 IDLE1.</p> <p>Selects the type of DS1 (Digital Signal-level 1) /DS3 (Digital Signal-level 3) alarm.</p> <p>AIS, selects Alarm Indication Signal (AIS) alarm when an unframed all-ones signal is received for DS1/DS3.</p> <p>RAI, selects Remote Alarm Indicates (RAI) DS1 (Digital Signal-level 1) which shows the following:</p> <p>For SF framing: The RAI alarm is declared when bit 2 in each timeslot contains "0".</p>

**:FETCh[1..n]:DATA:TELeom:DSN:ALARm:
DS[1..n]:CURRent?**

For ESF framing: The RAI alarm is declared when eight "ones" followed by eight "zeros" pattern is received continuously in the data link (FDL).

OOE, selects Out of Frame (OOE) error which indicates that four consecutive frame bit errors are detected for DS1(Digital Signal-level 1)/DS3 (Digital Signal-level 3).

RDI1, Remote Defect Indicator alarm is declared when both X-bits of the M-Frame are set to "0" for DS3 (Digital Signal-level 3).

IDLE1, Idle alarm is declared when subframe 3 of the M-frame contains zeros (0) for the three C-bits, ones (1) for X-bits, 1100... repeating sequence with the first two bits following each control bit set to 11 for the information bits.

Response Syntax <Current>

Response(s) Current:
The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the current status of Digital Signal-level 1 (DS1)/Digital Signal-level 3 (DS3) 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.

**:FETCh[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:CURRent?****Example(s)**

- * SOUR:DATA:TEL:DSN:ALAR:DS1:TYPE AIS
- * SOUR:DATA:TEL:DSN:ALAR:DS1 ON
- * FETC:DATA:TEL:DSN:ALAR:DS1:CURR? AIS

Returns the current alarm status.**Note**

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]:TYPE
- * SOURce[1..n]:DATA:TELEcom:DSN:ALARm:
DS[1..n]

:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR: DS[1..n]:AUTomated:TYPE

Description	<p>This command selects the type of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error for automated injection.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR: DS[1..n]:AUTomated:TYPE<wsp>CRC6 FBIT CBIT FBIT1 PBIT FEBE</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CRC6 FBIT CBIT FBIT1 PBIT FEBE.</p> <p>Selects the type of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error for automated injection.</p> <p>CRC6, selects the CRC-6 (Cyclical Redundancy Check-6) error which indicates that one or more bit errors have been detected in a block of data through cyclical redundancy check for DS1 (Digital Signal-level 1).</p> <p>FBIT, selects the F-Bit (Framing Bit) error which indicates that an incorrect value appeared in a bit position reserved for DS1 (Digital Signal-level 1) framing.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:ERROR:
DS[1..n]:AUTomated:TYPE**

CBIT, selects the CBIT (Control-Bit) error which indicates that the three C-bits reserved to control bit stuffing are different of "111" and "000" for DS3 (Digital Signal-level 3).

FBIT1, selects the DS3 (Digital Signal-level 3) Framing-Bit (FBIT1) error which indicates that the frame alignment pattern received is different for "1001".

PBIT, selects the Parity-Bit (PBIT) error which indicates that the P-Bits does not match the parity of all the information bits following the first X-Bit of the previous DS3 (Digital Signal-level 3) frame.

FEBE, selects the Far-End Block Error (FEBE) when the three FEBE bits reserved for DS3 (Digital Signal-level 3) framing or parity error detection contain the "000" pattern.

:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:TYPE

Example(s) * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:TYPE CRC6
 * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:TYPE?
 Returns CRC6

Note FTB/IQS-8140 Transport Blazer does not support
 this command.

See Also * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
 AUTomated:TYPE?
 * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
 AUTomated:RATE
 * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
 AUTomated

**:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR:
DS[1..n]:AUTomated:TYPE?**

Description	<p>This query returns the type of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error for automated injection.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR: DS[1..n]:AUTomated:TYPE?</code>
Parameter(s)	None
Response Syntax	<Error>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error for the automated injection.</p> <p>CRC6, Cyclic Redundancy Check (CRC)6 is selected as Digital Signal-level 1 (DS1) error.</p> <p>FBIT, Framing Bit (FBIT) is selected as Digital Signal-level 1 (DS1) error.</p> <p>CBIT, Control-Bit (C-BIT) is selected as Digital Signal-level 3 (DS3) error.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:AUTomated:TYPE?**

FBIT1, Framing bit (FBIT) is selected as Digital Signal-level 3(DS3) error.

PBIT, Parity bit (P-BIT) is selected as Digital Signal-level 3(DS3) error.

FEBE, Far-End Block Error (FEBE) is selected as Digital Signal-level 3(DS3) error.

Example(s)

* SOUR:DATA:TEL:DSN:ERR:DS1:AUT:TYPE CRC6
* SOUR:DATA:TEL:DSN:ERR:DS1:AUT:TYPE?
Returns CRC6

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
AUTomated:RATE
* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
AUTomated

**:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR:
DS[1..n]:AUTomated:RATE**

Description	<p>This command sets the injection rate for the selected DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR: DS[1..n]:AUTomated:RATE <wsp> <Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the injection rate for the selected DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.</p>

:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:RATE

Example(s) * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:TYPE CRC6
 * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:RATE
 1.0E-09
 * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:RATE?
 Returns 1.0E-09

Note FTB/IQS-8140 Transport Blazer does not support
 this command.

See Also * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
 AUTomated:TYPE
 * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
 AUTomated:RATE?
 * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
 AUTomated

**:SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:RATE?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injection rate will be returned.</p>
Response Syntax	<pre><Rate></pre>

:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected DS1(Digital Signal-level 1)/DS3 (Digital Signal-level 3) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:DSN:ERR:DS1:AUT:TYPE CRC6</p> <p>* SOUR:DATA:TEL:DSN:ERR:DS1:AUT:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:DSN:ERR:DS1:AUT:RATE? Returns 1.0E-09</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AUTomated</p>

**:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR:
DS[1..n]:AUTomated**

Description	<p>This command enables or disables the selected automated DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error at the rate specified or continuously.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR: DS[1..n]:AUTomated <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error injection.</p> <hr/>

:SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS[1..n]:AUTomated

Example(s)

- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:TYPE CRC6
- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:RATE
1.0E-09
- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT ON
- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT? Returns 1

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AUTomated:TYPE
- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AUTomated:RATE
- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:AUTomated?**

Description	<p>This query returns the status of automated DS1(Digital Signal-level 1)/DS3 (Digital Signal-level 3) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated DS1(Digital Signal-level 1)/DS3 (Digital Signal-level 3) error injection.</p>

:SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS[1..n]:AUTomated?

Example(s)

- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:TYPE CRC6
- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:RATE 1.0E-09
- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT ON
- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT? Returns 1

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AUTomated:TYPE
- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AUTomated:RATE
- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR:
DS[1..n]:AUTomated:CONTInuous**

Description	<p>This command enables or disables the continuous rate of automated DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:DSN:ERROR: DS[1..n]:AUTomated:CONTInuous <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error injection continuously.</p> <hr/>

:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:CONTInuous

Example(s)

- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:TYPE CRC6
- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:CONT ON
- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:CONT?

Returns 1

- * SOUR:DATA:TEL:DSN:ERR:DS1:AUT ON

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
AUTomated:TYPE
- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
AUTomated
- * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
AUTomated:CONTInuous?

**:SOURce[1..n]:DATA:TELEcom:DSN:ERRor:
DS[1..n]:AUTomated:CONTInuous?**

Description	<p>This query returns the status of continuous rate of automated DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:CONTInuous?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuous rate of automated DS1 (Digital Signal-level 1)/DS3 (Digital Signal-level 3) error injection.</p>

:SOURce[1..n]:DATA:TELEcom:DSN:ERRor: DS[1..n]:AUTomated:CONTInuous?

Example(s) * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:TYPE CRC6
 * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:CONT ON
 * SOUR:DATA:TEL:DSN:ERR:DS1:AUT:CONT?
 Returns 1
 * SOUR:DATA:TEL:DSN:ERR:DS1:AUT ON

Note FTB/IQS-8140 Transport Blazer does not support
 this query.

See Also * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
 AUTomated:TYPE
 * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
 AUTomated
 * SOURce[1..n]:DATA:TELEcom:DSN:ERRor:DS1:
 AUTomated:CONTInuous

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC

Description	<p>This command enables or disables the FEAC (Far-End Alarm And Control) configuration for the transmitter.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the FEAC configuration for the transmitter.</p>
Example(s)	<pre>* SOUR:DATA:TEL:DSN:FEAC ON * SOUR:DATA:TEL:DSN:FEAC? Returns 1</pre>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this command.</p>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:DSN:FEAC?</pre>

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC?

Description	<p>This query returns the status of FEAC (Far-End Alarm And Control) configuration for the transmitter.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:DSN:FEAC?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of FEAC configuration for the transmitter.</p>
Example(s)	<p>* SOUR:DATA:TEL:DSN:FEAC ON</p> <p>* SOUR:DATA:TEL:DSN:FEAC? Returns 1</p>
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:DSN:FEAC

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:CODeword

Description This command selects the manual codeword for Alarm/Status Unassigned Messages.

At *RST, this value is set to DS3IR.

Syntax :SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:CODeword <wsp> DS3EFSA | DS3LOS |
DS3OOF | DS3AR | DS3IR | DS3EFNSA | CEFailure |
MDS1LOS | DS1EFSA | SDS1LOS | DS1EFNSA |
U00000010 | U00000100 | U00001000 | U00001100 |
U00010000 | U00010010 | U00010100 | U00010110 |
U00011000 | U00011010 | U00100000 | U00100010 |
U00100100 | U00101000 | U00101110 | U00110000 |
U00111110 | U01000000 | U01111010 | U01111100 |
U01111110 | IVALue

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:CODeword**

Parameter(s)

Codeword:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

DS3EFSA|DS3LOS|DS3OOF|DS3AR|DS3IR|
DS3EFNSA|CEFailure|MDS1LOS|DS1EFSA|
SDS1LOS|DS1EFNSA|U00000010|U00000100|
U00001000|U00001100|U00010000|U00010010|
U00010100|U00010110|U00011000|U00011010|
U00100000|U00100010|U00100100|U00101000|
U00101110|U00110000|U00111110|U01000000|
U01111010|U01111100|U01111110|IVALue

Selects the manual codeword.

DS3EFSA, selects DS3 Eqpt. Failure (SA) (00110010) as manual code word.

DS3LOS, selects DS3 LOS (00011100).

DS3OOF, selects DS3 Out of Frame (00000000).

DS3AR, selects DS3 AIS Received (00101100).

DS3IR, selects DS3 IDLE Received (00110100).

DS3EFNSA, selects DS3 Eqpt. Failure (NSA) (00011110).

CEFailure, selects Common Eqpt. Failure (NSA) (00111010).

MDS1LOS, selects Multiple DS1 LOS (00101010).

DS1EFSA, selects DS1 Eqpt. Failure (SA) (00001010).

SDS1LOS, selects Single DS1 LOS (00111100).

DS1EFNSA, selects DS1 Eqpt. Failure (NSA) (00000110).

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:CODeword**

U00000010, selects Unassigned (00000010).
U00000100, selects Unassigned (00000100).
U00001000, selects Unassigned (00001000).
U00001100, selects Unassigned (00001100).
U00010000, selects Unassigned (00010000).
U00010010, selects Unassigned (00010010).
U00010100, selects Unassigned (00010100).
U00010110, selects Unassigned (00010110).
U00011000, selects Unassigned (00011000).
U00011010, selects Unassigned (00011010).
U00100000, selects Unassigned (00100000).
U00100010, selects Unassigned (00100010).
U00100100, selects Unassigned (00100100).
U00101000, selects Unassigned (00101000).
U00101110, selects Unassigned (00101110).
U00110000, selects Unassigned (00110000).
U00111110, selects Unassigned (00111110).
U01000000, selects Unassigned (01000000).
U01111010, selects Unassigned (01111010).
U01111100, selects Unassigned (01111100).
U01111110, selects Unassigned (01111110).
IVALue, selects Invalid Value (IVALUE).

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:CODeword

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:FEAC ON* SOUR:DATA:TEL:DSN:FEAC:MAN:COD DS3LOS* SOUR:DATA:TEL:DSN:FEAC:MAN:COD? Returns DS3LOS
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:FEAC* SOURce[1..n]:DATA:TELEcom:DSN:FEAC:MANual:CODEword?* SOURce[1..n]:DATA:TELEcom:DSN:FEAC:MANual:AMOUNT* SOURce[1..n]:DATA:TELEcom:DSN:FEAC:MANual:INJECT

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:CODeword?**

Description	<p>This query returns the manual codeword for Alarm/Status Unassigned Messages.</p> <p>At *RST, this value is set to DS3IR.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:CODeword?
Parameter(s)	None
Response Syntax	<Codeword>
Response(s)	<p>Codeword:</p> <p>The response data syntax for <Codeword> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the manual codeword.</p> <p>DS3EFSA, DS3 Eqpt. Failure (SA) (00110010) is selected as manual code word.</p> <p>DS3LOS, DS3 LOS (00011100) is selected.</p> <p>DS3OOF, DS3 Out of Frame (00000000) is selected.</p> <p>DS3AR, DS3 AIS Received (00101100) is selected.</p> <p>DS3IR, DS3 IDLE Received (00110100) is selected.</p> <p>DS3EFNSA, DS3 Eqpt. Failure (NSA) (00011110) is selected.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:CODeword?**

CEFailure, Common Eqpt. Failure (NSA) (00111010) is selected.
MDS1LOS, Multiple DS1 LOS (00101010) is selected.
DS1EFSA, DS1 Eqpt. Failure (SA) (00001010) is selected.
SDS1LOS, Single DS1 LOS (00111100) is selected.
DS1EFNSA, DS1 Eqpt. Failure (NSA) (00000110) is selected.
U00000010, Unassigned (00000010) is selected.
U00000100, Unassigned (00000100) is selected.
U00001000, Unassigned (00001000) is selected.
U00001100, Unassigned (00001100) is selected.
U00010000, Unassigned (00010000) is selected.
U00010010, Unassigned (00010010) is selected.
U00010100, Unassigned (00010100) is selected.
U00010110, Unassigned (00010110) is selected.
U00011000, Unassigned (00011000) is selected.
U00011010, Unassigned (00011010) is selected.
U00100000, Unassigned (00100000) is selected.
U00100010, Unassigned (00100010) is selected.
U00100100, Unassigned (00100100) is selected.
U00101000, Unassigned (00101000) is selected.
U00101110, Unassigned (00101110) is selected.
U00110000, Unassigned (00110000) is selected.
U00111110, Unassigned (00111110) is selected.
U01000000, Unassigned (01000000) is selected.
U01111010, Unassigned (01111010) is selected.
U01111100, Unassigned (01111100) is selected.
U01111110, Unassigned (01111110) is selected.
IVALue, Invalid Value (IVALUE) is selected.

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:CODeword?**

Example(s)	* SOUR:DATA:TEL:DSN:FEAC ON * SOUR:DATA:TEL:DSN:FEAC:MAN:COD DS3LOS * SOUR:DATA:TEL:DSN:FEAC:MAN:COD? Returns DS3LOS
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:DSN:FEAC * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:CODEword * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:AMOUNT * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:INJECT

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:AMOut**

Description	<p>This command sets the amount for Alarm/Status Unassigned Messages.</p> <p>At *RST, this value is set to 10.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:AMOut<wsp><Amount> MAXimum MINimum</p>
Parameter(s)	<p>Amount: The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the amount for Alarm/Status Unassigned Messages. Choices are 1 through 15.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:AMOUNT**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:FEAC ON* SOUR:DATA:TEL:DSN:FEAC:MAN:COD DS3LOS* SOUR:DATA:TEL:DSN:FEAC:MAN:AMO 10* SOUR:DATA:TEL:DSN:FEAC:MAN:AMO? Returns 10
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:FEAC* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:CODEword* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:AMOUNT?* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:INJect

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:AMOUNT?**

Description	<p>This query returns the amount for Alarm/Status Unassigned Messages.</p> <p>At *RST, this value is set to 10.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual :AMOUNT? [<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount will be returned.</p>
Response Syntax	<p><Amount></p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:AMOut?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount for Alarm/Status Unassigned Messages.
Example(s)	* SOUR:DATA:TEL:DSN:FEAC ON * SOUR:DATA:TEL:DSN:FEAC:MAN:COD DS3LOS * SOUR:DATA:TEL:DSN:FEAC:MAN:AMO 10 * SOUR:DATA:TEL:DSN:FEAC:MAN:AMO? Returns 10
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:DSN:FEAC * SOURce[1..n]:DATA:TELEcom:DSN:FEAC:MANual:CODEword * SOURce[1..n]:DATA:TELEcom:DSN:FEAC:MANual:AMOut * SOURce[1..n]:DATA:TELEcom:DSN:FEAC:MANual:INJect

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
MANual:INJect**

Description	<p>This command injects the Alarm/Status Unassigned Messages.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:INJect</p>
Parameter(s)	<p>None</p>
Example(s)	<p>* SOUR:DATA:TEL:DSN:FEAC ON * SOUR:DATA:TEL:DSN:FEAC:MAN:COD DS3LOS * SOUR:DATA:TEL:DSN:FEAC:MAN:AMO 10 * SOUR:DATA:TEL:DSN:FEAC:MAN:AMO? Returns 10 * SOUR:DATA:TEL:DSN:FEAC:MAN:INJ</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this command.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:DSN:FEAC * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:CODEword * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: MANual:AMOUNT</p>

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous:CODeword

Description

This command selects the continuous codeword for Alarm/Status Unassigned Messages.

At *RST, this value is set to DS3IR.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTInous:CODeword<wsp>DS3EFSA|
DS3LOS|DS3OOF|DS3AR|DS3IR|DS3EFNSA|
CEFailure|MDS1LOS|DS1EFSA|SDS1LOS|
DS1EFNSA|U00000010|U00000100|U00001000|
U00001100|U00010000|U00010010|U00010100|
U00010110|U00011000|U00011010|U00100000|
U00100010|U00100100|U00101000|U00101110|
U00110000|U00111110|U01000000|U01111010|
U01111100|U01111110|IVALue
```

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTinuous:CODeword**

Parameter(s)

Codeword:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

DS3EFSA|DS3LOS|DS3OOF|DS3AR|DS3IR|
DS3EFNSA|CEFailure|MDS1LOS|DS1EFSA|
SDS1LOS|DS1EFNSA|U00000010|U00000100|
U00001000|U00001100|U00010000|U00010010|
U00010100|U00010110|U00011000|U00011010|
U00100000|U00100010|U00100100|U00101000|
U00101110|U00110000|U00111110|U01000000|
U01111010|U01111100|U01111110|IVALue

Selects the continuous codeword.

DS3EFSA, selects DS3 Eqpt. Failure (SA) (00110010) as continuous code word.

DS3LOS, selects DS3 LOS (00011100).

DS3OOF, selects DS3 Out of Frame (00000000).

DS3AR, selects DS3 AIS Received (00101100).

DS3IR, selects DS3 IDLE Received (00110100).

DS3EFNSA, selects DS3 Eqpt. Failure (NSA) (00011110).

CEFailure, selects Common Eqpt. Failure (NSA) (00111010).

MDS1LOS, selects Multiple DS1 LOS (00101010).

DS1EFSA, selects DS1 Eqpt. Failure (SA) (00001010).

SDS1LOS, selects Single DS1 LOS (00111100).

DS1EFNSA, selects DS1 Eqpt. Failure (NSA) (00000110).

U00000010, selects Unassigned (00000010).

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTInous:CODeword**

U00000100, selects Unassigned (00000100).
U00001000, selects Unassigned (00001000).
U00001100, selects Unassigned (00001100).
U00010000, selects Unassigned (00010000).
U00010010, selects Unassigned (00010010).
U00010100, selects Unassigned (00010100).
U00010110, selects Unassigned (00010110).
U00011000, selects Unassigned (00011000).
U00011010, selects Unassigned (00011010).
U00100000, selects Unassigned (00100000).
U00100010, selects Unassigned (00100010).
U00100100, selects Unassigned (00100100).
U00101000, selects Unassigned (00101000).
U00101110, selects Unassigned (00101110).
U00110000, selects Unassigned (00110000).
U00111110, selects Unassigned (00111110).
U01000000, selects Unassigned (01000000).
U01111010, selects Unassigned (01111010).
U01111100, selects Unassigned (01111100).
U01111110, selects Unassigned (01111110).
IVALue, selects Invalid Value.

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous:CODeword

Example(s)

- * SOUR:DATA:TEL:DSN:FEAC ON
- * SOUR:DATA:TEL:DSN:FEAC:CONT:COD
DS3LOS
- * SOUR:DATA:TEL:DSN:FEAC:CONT:COD?
Returns DS3LOS

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:DSN:FEAC
- * SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTInous:CODEword?

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTInous:CODeword?**

Description	This query returns the continuous codeword for Alarm/Status Unassigned Messages. At *RST, this value is set to DS3IR.
Syntax	:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous:CODeword?
Parameter(s)	None
Response Syntax	<Codeword>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTInous:CODeword?**

Response(s)

Codeword:

The response data syntax for <Codeword> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the continuous codeword.

DS3EFSA, DS3 Eqpt. Failure (SA) (00110010) is selected as continuous code word.

DS3LOS, DS3 LOS (00011100) is selected.

DS3OOF, DS3 Out of Frame (00000000) is selected.

DS3AR, DS3 AIS Received (00101100) is selected.

DS3IR, DS3 IDLE Received (00110100) is selected.

DS3EFNSA, DS3 Eqpt. Failure (NSA) (00011110) is selected.

CEFailure, Common Eqpt. Failure (NSA) (00111010) is selected.

MDS1LOS, Multiple DS1 LOS (00101010) is selected.

DS1EFSA, DS1 Eqpt. Failure (SA) (00001010) is selected.

SDS1LOS, Single DS1 LOS (00111100) is selected.

DS1EFNSA, DS1 Eqpt. Failure (NSA) (00000110) is selected.

U00000010, Unassigned (00000010) is selected.

U00000100, Unassigned (00000100) is selected.

U00001000, Unassigned (00001000) is selected.

U00001100, Unassigned (00001100) is selected.

U00010000, Unassigned (00010000) is selected.

U00010010, Unassigned (00010010) is selected.

U00010100, Unassigned (00010100) is selected.

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous:CODeword?

U00010110, Unassigned (00010110) is selected.
 U00011000, Unassigned (00011000) is selected.
 U00011010, Unassigned (00011010) is selected.
 U00100000, Unassigned (00100000) is selected.
 U00100010, Unassigned (00100010) is selected.
 U00100100, Unassigned (00100100) is selected.
 U00101000, Unassigned (00101000) is selected.
 U00101110, Unassigned (00101110) is selected.
 U00110000, Unassigned (00110000) is selected.
 U00111110, Unassigned (00111110) is selected.
 U01000000, Unassigned (01000000) is selected.
 U01111010, Unassigned (01111010) is selected.
 U01111100, Unassigned (01111100) is selected.
 U01111110, Unassigned (01111110) is selected.
 IVALue, Invalid Value is selected.

Example(s)

```
* SOUR:DATA:TEL:DSN:FEAC ON
* SOUR:DATA:TEL:DSN:FEAC:CONT:COD
DS3LOS
* SOUR:DATA:TEL:DSN:FEAC:CONT:COD?
Returns DS3LOS
```

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

```
* SOURce[1..n]:DATA:TELEcom:DSN:FEAC
* SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTInous:CODEword
```

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous

Description	<p>This command enables or disables the Continuous Codeword feature.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous <wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the Continuous Codeword feature.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTInous**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:FEAC ON* SOUR:DATA:TEL:DSN:FEAC:CONT:COD DS3LOS* SOUR:DATA:TEL:DSN:FEAC:CONT ON* SOUR:DATA:TEL:DSN:FEAC:CONT? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:FEAC* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous:CODEword* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous?

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous?

Description	<p>This query returns the status of the Continuous Codeword feature.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: CONTInous?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Continuous Codeword feature.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTInous?****Example(s)**

- * SOUR:DATA:TEL:DSN:FEAC ON
- * SOUR:DATA:TEL:DSN:FEAC:CONT:COD
DS3LOS
- * SOUR:DATA:TEL:DSN:FEAC:CONT ON
- * SOUR:DATA:TEL:DSN:FEAC:CONT? Returns 1

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:DSN:FEAC
 - * SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTInous:CODEword
 - * SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
CONTInous
-

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:CODeword

Description	<p>This command selects the control codeword for Loopback commands.</p> <p>At *RST, this value is set to LLACTivate.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:CODeword <wsp> LLACTivate LLDeactivate</pre>
Parameter(s)	<p>Codeword:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LLACTivate LLDeactivate.</p> <p>Selects the control codeword for loopback commands.</p> <p>LLACTivate, selects Line Loopback Activate (00001110) as control codeword.</p> <p>LLDeactivate, selects Line Loopback Deactivate (00111000) as control codeword.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol:CODeword****Example(s)**

* SOUR:DATA:TEL:DSN:FEAC ON
* SOUR:DATA:TEL:DSN:FEAC:LOOP:CONT:COD
LLAC
* SOUR:DATA:TEL:DSN:FEAC:LOOP:CONT:COD?
Returns LLACTIVATE

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:DSN:FEAC
* SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol:CODEword?
* SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol:AMOUNT
* SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:INJect

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:CODeword?

Description	<p>This query returns the control codeword for Loopback commands.</p> <p>At *RST, this value is set to LLACTivate.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:CODeword?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Codeword></p>
Response(s)	<p>Codeword: The response data syntax for <Codeword> is defined as a <CHARACTER RESPONSE DATA> element. Returns the control codeword for loopback commands. LLACTivate, Line Loopback Activate (00001110) is selected as control codeword. LLDeactivate, Line Loopback Deactivate (00111000) is selected as control codeword.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol:CODeword?****Example(s)**

* SOUR:DATA:TEL:DSN:FEAC ON
* SOUR:DATA:TEL:DSN:FEAC:LOOP:CONT:COD
LLAC
* SOUR:DATA:TEL:DSN:FEAC:LOOP:CONT:COD?
Returns LLACTIVATE

Note

FTB/IQS-8140 Transport Blazer does not support
this query.

See Also

* SOURce[1..n]:DATA:TELEcom:DSN:FEAC
* SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol:CODEword
* SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol:AMOUNT
* SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:INJect

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol:AMOut**

Description	<p>This command sets the amount of control codeword for Loopback commands.</p> <p>At *RST, this value is set to 10.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:AMOut <wsp> <Amount> MAXimum MINimum</pre>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the amount of control codeword for Loopback commands.</p> <p>Choices are 1 through 15.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol:AMOut**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:FEAC ON* SOUR:DATA:TEL:DSN:FEAC:LOOP:CONT:COD LLAC* SOUR:DATA:TEL:DSN:FEAC:LOOP:CONT:AMO 10* SOUR:DATA:TEL:DSN:FEAC:LOOP:CONT:AMO? Returns 10
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:FEAC* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:CODEword* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:AMOut?* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:INJect

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol:AMOut?**

Description	<p>This query returns the amount of control codeword for Loopback commands.</p> <p>At *RST, this value is set to 10.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:AMOut?[<wsp> MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount will be returned.</p>
Response Syntax	<p><Amount></p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol:AMOUNT?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of control codeword for Loopback commands.
Example(s)	* SOUR:DATA:TEL:DSN:FEAC ON * SOUR:DATA:TEL:DSN:FEAC:LOOP:CONT:COD LLAC * SOUR:DATA:TEL:DSN:FEAC:LOOP:CONT:AMO 10 * SOUR:DATA:TEL:DSN:FEAC:LOOP:CONT:AMO? Returns 10
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:DSN:FEAC * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:CODEword * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol:AMOUNT * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:INJect

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:CODeword

Description This command selects the channel codeword for Loopback commands.

At *RST, this value is set to DS3L.

Syntax :SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback :CHANnel:CODeword <wsp> DS3L
|DS1L1|DS1L2|DS1L3|DS1L4|DS1L5|DS1L6|
DS1L7|DS1L8|DS1L9|DS1L10|DS1L11|DS1L12|
DS1L13|DS1L14|DS1L15|DS1L16|DS1L17|
DS1L18|DS1L19|DS1L20|DS1L21|DS1L22|
DS1L23|DS1L24|DS1L25|DS1L26|DS1L27|
DS1L28|DS1LALL

Parameter(s) Codeword:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
DS3L|DS1L1|DS1L2|DS1L3|DS1L4|DS1L5|
DS1L6|DS1L7|DS1L8|DS1L9|DS1L10|DS1L11|
DS1L12|DS1L13|DS1L14|DS1L15|DS1L16|
DS1L17|DS1L18|DS1L19|DS1L20|DS1L21|
DS1L22|DS1L23|DS1L24|DS1L25|DS1L26|
DS1L27|DS1L28|DS1LALL
Selects the channel codeword for Loopback commands.

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CHANnel:CODeword**

DS3L, selects DS3 Line (00110110) as channel codeword.

DS1L1, selects DS1 Line-No1 (01000010).

DS1L2, selects DS1 Line-No2 (01000100).

DS1L3, selects DS1 Line-No3 (01000110).

DS1L4, selects DS1 Line-No4 (01001000).

DS1L5, selects DS1 Line-No5 (01001010).

DS1L6, selects DS1 Line-No6 (01001100).

DS1L7, selects DS1 Line-No7 (01001110).

DS1L8, selects DS1 Line-No8 (01010000).

DS1L9, selects DS1 Line-No9 (01010010).

DS1L10, selects DS1 Line-No10 (01010100).

DS1L11, selects DS1 Line-No11 (01010110).

DS1L12, selects DS1 Line-No12 (01011000).

DS1L13, selects DS1 Line-No13 (01011010).

DS1L14, selects DS1 Line-No14 (01011100).

DS1L15, selects DS1 Line-No15 (01011110).

DS1L16, selects DS1 Line-No16 (01100000).

DS1L17, selects DS1 Line-No17 (01100010).

DS1L18, selects DS1 Line-No18 (01100100).

DS1L19, selects DS1 Line-No19 (01100110).

DS1L20, selects DS1 Line-No20 (01101000).

DS1L21, selects DS1 Line-No21 (01101010).

DS1L22, selects DS1 Line-No22 (01101100).

DS1L23, selects DS1 Line-No23 (01101110).

DS1L24, selects DS1 Line-No24 (01110000).

DS1L25, selects DS1 Line-No25 (01110010).

DS1L26, selects DS1 Line-No26 (01110100).

DS1L27, selects DS1 Line-No27 (01110110).

DS1L28, selects DS1 Line-No28 (01111000).

DS1LALL, selects DS1 Line-All (00100110).

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:CODeword

Example(s)

- * SOUR:DATA:TEL:DSN:FEAC ON
- * SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:COD DS3L2
- * SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:COD? Returns DS3L2

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:DSN:FEAC
- * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:CODEword?
- * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:AMOUNT
- * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:INJect

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CHANnel:CODeword?**

Description	<p>This query returns the channel codeword for Loopback commands.</p> <p>At *RST, this value is set to DS3L.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback :CHANnel:CODeword?
Parameter(s)	None
Response Syntax	<Codeword>
Response(s)	<p>Codeword:</p> <p>The response data syntax for <Codeword> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the channel codeword for Loopback Commands.</p> <p>DS3L, DS3 Line (00110110) is selected as channel codeword.</p> <p>DS1L1, DS1 Line-No1 (01000010) is selected.</p> <p>DS1L2, DS1 Line-No2 (01000100) is selected.</p> <p>DS1L3, DS1 Line-No3 (01000110) is selected.</p> <p>DS1L4, DS1 Line-No4 (01001000) is selected.</p> <p>DS1L5, DS1 Line-No5 (01001010) is selected.</p> <p>DS1L6, DS1 Line-No6 (01001100) is selected.</p> <p>DS1L7, DS1 Line-No7 (01001110) is selected.</p> <p>DS1L8, DS1 Line-No8 (01010000) is selected.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CHANnel:CODeword?**

DS1L9, DS1 Line-No9 (01010010) is selected.
DS1L10, DS1 Line-No10 (01010100) is selected.
DS1L11, DS1 Line-No11 (01010110) is selected.
DS1L12, DS1 Line-No12 (01011000) is selected.
DS1L13, DS1 Line-No13 (01011010) is selected.
DS1L14, DS1 Line-No14 (01011100) is selected.
DS1L15, DS1 Line-No15 (01011110) is selected.
DS1L16, DS1 Line-No16 (01100000) is selected.
DS1L17, DS1 Line-No17 (01100010) is selected.
DS1L18, DS1 Line-No18 (01100100) is selected.
DS1L19, DS1 Line-No19 (01100110) is selected.
DS1L20, DS1 Line-No20 (01101000) is selected.
DS1L21, DS1 Line-No21 (01101010) is selected.
DS1L22, DS1 Line-No22 (01101100) is selected.
DS1L23, DS1 Line-No23 (01101110) is selected.
DS1L24, DS1 Line-No24 (01110000) is selected.
DS1L25, DS1 Line-No25 (01110010) is selected.
DS1L26, DS1 Line-No26 (01110100) is selected.
DS1L27, DS1 Line-No27 (01110110) is selected.
DS1L28, DS1 Line-No28 (01111000) is selected.
DS1LALL, DS1 Line-All (00100110) is selected.

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CHANnel:CODeword?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:FEAC ON* SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:COD DS3L2* SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:COD? Returns DS3L2
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:FEAC* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:CODEword* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:AMOUNT* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:INJect

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CHANnel:AMOut**

Description	<p>This command sets the amount of channel codeword for Loopback commands.</p> <p>At *RST, this value is set to 10.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:AMOut<wsp> <Amount> MAXimum MINimum</p>
Parameter(s)	<p>Amount: The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the amount of channel codeword for Loopback commands. Choices are 1 through 15.</p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CHANnel:AMOUNT**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:DSN:FEAC ON* SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:COD DS3L2* SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:AMO 10* SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:AMO? Returns 10
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:DSN:FEAC* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:CODEword* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:AMOUNT?* SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:INJect

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CHANnel:AMOUNT?**

Description	<p>This query returns the amount of channel codeword for Loopback commands.</p> <p>At *RST, this value is set to 10.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:AMOUNT? [<wsp> MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount will be returned.</p>
Response Syntax	<p><Amount></p>

**:SOURce[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CHANnel:AMOUNT?**

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of channel codeword for Loopback commands.</p>
Example(s)	<pre>* SOUR:DATA:TEL:DSN:FEAC ON * SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:COD DS3L2 * SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:AMO 10 * SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:AMO? Returns 10</pre>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:DSN:FEAC * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:CODEword * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:AMOUNT * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:INJect</pre>

:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:INJect

Description	<p>This command injects the Loopback commands.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:INJect</pre>
Parameter(s)	None
Example(s)	<pre>* SOUR:DATA:TEL:DSN:FEAC ON * SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:COD DS3L2 * SOUR:DATA:TEL:DSN:FEAC:LOOP:CHAN:AMO 10 * SOUR:DATA:TEL:DSN:FEAC:LOOP:INJ</pre>
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:DSN:FEAC * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:CODEword * SOURce[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel:AMOUNT</pre>

:SENSe[1..n]:DATA:TELEcom:DSN:FEAC

Description	<p>This command enables or disables the FEAC (Far-End Alarm And Control) configuration for the receiver.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:DSN:FEAC <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the FEAC configuration for the receiver.</p>
Example(s)	<pre>* SENS:DATA:TEL:DSN:FEAC ON * SENS:DATA:TEL:DSN:FEAC? Returns 1</pre>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this command.</p>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:DSN:FEAC?</pre>

:SENSe[1..n]:DATA:TELEcom:DSN:FEAC?

Description	This query returns the status of FEAC (Far-End Alarm And Control) configuration for the receiver. At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:DSN:FEAC?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the FEAC configuration for the receiver.
Example(s)	* SENS:DATA:TEL:DSN:FEAC ON * SENS:DATA:TEL:DSN:FEAC? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SENSe[1..n]:DATA:TELEcom:DSN:FEAC

**:FETCh[1..n]:DATA:TELEcom:DSN:FEAC:
MESSAge?**

Description	<p>This query returns the Alarm/Status Unassigned Messages.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:DSN:FEAC: MESSAge? <wsp>CURRent PREVious</p>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CURRent PREVious.</p> <p>Selects the Alarm/Status Unassigned Messages. CURRent, selects Alarm/Status Unassigned Messages for Current state. PREVious, selects Alarm/Status Unassigned Messages for Previous state.</p>
Response Syntax	<p><Message></p>

:FETCh[1..n]:DATA:TELEcom:DSN:FEAC: MESSAge?

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the Alarm/Status Unassigned Messages.
Example(s)	* FETC:DATA:TEL:DSN:FEAC:MESS? CURR Returns the current Alarm/Status Unassigned Messages.
Note	FTB/IQS-8140 Transport Blazer does not support this query.

**:FETCh[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CONTrol?**

Description	<p>This query returns the control Loopback commands for Current or Previous state.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol? <wsp> CURRent PREVious</p>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CURRent PREVious.</p> <p>Selects the control loopback commands. CURRent, sets the control loopback commands for Current state. PREVious, sets the control loopback commands for Previous state.</p>
Response Syntax	<p><Command></p>
Response(s)	<p>Command:</p> <p>The response data syntax for <Command> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the Alarm/Status Unassigned Messages.</p>

:FETCh[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CONTrol?

Example(s)	* FETC:DATA:TEL:DSN:FEAC:LOOP:CONT? CURR Returns control loopback commands.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* FETCh[1..n]:DATA:TELEcom:DSN:FEAC: CHANnel?

**:FETCh[1..n]:DATA:TELEcom:DSN:FEAC:
LOOPback:CHANnel?**

Description	<p>This query returns the channel Loopback commands for Current or Previous state.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel?<wsp>CURRent PREVious</p>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CURRent PREVious.</p> <p>Selects the channel loopback commands.</p> <p>CURRent, sets the channel loopback commands for Current state.</p> <p>PREVious, sets the channel loopback commands for Previous state.</p>
Response Syntax	<p><Command></p>

:FETCh[1..n]:DATA:TELEcom:DSN:FEAC: LOOPback:CHANnel?

Response(s)	Command: The response data syntax for <Command> is defined as a <STRING RESPONSE DATA> element. Returns the channel Loopback commands for Current or Previous state.
Example(s)	* FETC:DATA:TEL:DSN:FEAC:LOOP:CHAN? CURR Returns the channel loopback commands.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* FETCh[1..n]:DATA:TELEcom:DSN:FEAC:CONTrol?

:FETCh[1..n]:DATA:TELEcom:DSN:FEAC:LINK?

Description	<p>This query returns the activity for the No Activity (All 1's), Alarm/Status, Loopback and Unassigned parameters during the last second of measurement.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:DSN:FEAC:LINK? <wsp>NOACTivity ASTatus LOOPback UNASSigned</pre>
Parameter(s)	<p>Activity:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NOACTivity ASTatus LOOPback UNASSigned.</p> <p>Selects the activity for the following parameters during the last second of measurement.</p> <p>NOACTivity, selects No Activity (All 1's).</p> <p>ASTatus, selects Alarm/Status.</p> <p>LOOPback, selects Loopback.</p> <p>UNASSigned, selects Unassigned.</p>
Response Syntax	<pre><Status></pre>

:FETCh[1..n]:DATA:TELecom:DSN:FEAC:LINK?

Response(s)	Status: The response data syntax for the parameter is defined as a <STRING RESPONSE DATA> element. Returns the activity for the selected parameters during the last second of measurement.
Example(s)	* FETC:DATA:TEL:DSN:FEAC:LINK? LOOP Returns the activity for the selected parameters during the last second of measurement.
Note	FTB/IQS-8140 Transport Blazer does not support this query.

**:FETCh[1..n]:DATA:TELEcom:DSN:DS[1..n]:PM:
STATistics?**

Description	<p>This query returns the performance monitoring statistics of Digital Signal levels.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:DSN:DS[1..n]:PM: STATistics?<wsp>G826ISM M2100ISM,EFS EB ES SES BBE UAS ESR SESR BBER,NEND FEND</pre>
Parameter(s)	<p>Standard:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: G826ISM M2100ISM.</p> <p>Selects the performance monitoring standard number.</p> <p>G826ISM, selects G.826 ISM as a standard number.</p> <p>M2100ISM, selects M.2100 ISM as a standard number.</p>

:FETCh[1..n]:DATA:TELecom:DSN:DS[1..n]:PM: STATistics?

Type:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

EFS|EB|ES|SES|BBE|UAS|ESR|SESR|BBER.

Selects the performance monitoring statistics.

EFS, selects Error Free Seconds (EFS).

EB, selects Errored Block (EB).

ES, selects Errored Seconds (ES).

SES, selects Severely Errored Seconds (SES).

BBE, selects Background Block Error (SES).

UAS, selects Unavailable Second (UAS).

ESR, selects Errored Second Ratio (ESR).

SESR, selects Severely Errored Second Ratio (SESR).

BBER, selects Background Block Error Ratio (BBER).

**:FETCh[1..n]:DATA:TELEcom:DSN:DS[1..n]:PM:
STATistics?**

End:

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

NEND|FEND.

Selects Near-End or Far-End.

NEND, selects the standard for Near-End.

FEND, selects the standard for Far-End.

Response Syntax

<Statistics>

Response(s)

Statistics:

The response data syntax for <Statistics> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the performance monitoring statistics of Digital Signal (DS).

Example(s)

* FETC:DATA:TEL:DSN:DS1:PM:STAT?
G826ISM, EFS, NEND

Returns the performance monitoring statistics of DS1.

PDH Command Reference

:SOURce[1..n]:DATA:TELEcom:PDH:E:ENABLEd

Description This command enables or disables the activation of European standard for digital transmission -level 0 (E0)/64K testing.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:PDH:E:ENABLEd
<wsp> <Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the activation of European standard for digital transmission-level 0 (E0)/64K testing.

Example(s) * SOUR:DATA:TEL:PDH:E:ENAB ON
* SOUR:DATA:TEL:PDH:E:ENAB? Returns 1

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also * SOURce[1..n]:DATA:TELEcom:PDH:E:ENABLEd?

:SOURce[1..n]:DATA:TELEcom:PDH:E:ENABLEd?

Description	This query returns the status of European standard for digital transmission-level 0 (E0)/64K testing. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:PDH:E:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of European standard for digital transmission-level 0 (E0)/64K testing.
Example(s)	* SOUR:DATA:TEL:PDH:E:ENAB ON * SOUR:DATA:TEL:PDH:E:ENAB? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABLEd

:SENSe[1..n]:DATA:TELEcom:PDH:E:ENABLEd

Description This command enables or disables the activation of European standard for digital transmission-level 0 (E0)/64K testing.

At *RST, this value is set to OFF.

Syntax :SENSe[1..n]:DATA:TELEcom:PDH:E:ENABLEd
<wsp> <Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the activation of European standard for digital transmission-level 0 (E0)/64K testing.

Example(s) * SENS:DATA:TEL:PDH:E:ENAB ON
* SENS:DATA:TEL:PDH:E:ENAB? Returns 1

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also * SENSe[1..n]:DATA:TELEcom:PDH:E:ENABLEd?

:SENSe[1..n]:DATA:TELEcom:PDH:E:ENABLEd?

Description	This query returns the status of European standard for digital transmission-level 0 (E0)/64K testing. At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:PDH:E:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for the parameter is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of European standard for digital transmission-level 0 (E0)/64K testing.
Example(s)	* SENS:DATA:TEL:PDH:E:ENAB ON * SENS:DATA:TEL:PDH:E:ENAB? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SENSe[1..n]:DATA:TELEcom:PDH:E:ENABLEd

:SOURce[1..n]:DATA:TELEcom:PDH:E:MODE

Description This command selects the channel timeslot data rate for the transmitter.

At *RST, this value is set to E64K.

Syntax :SOURce[1..n]:DATA:TELEcom:PDH:E:MODE
<wsp>E64K|E56K

Parameter(s) Mode:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: E64K|E56K.
Selects the channel timeslot data rate for the transmitter.
E64K, selects a timeslot data rate of 64 Kbps uses 8 bits to carry the payload information.
E56K, selects a timeslot data rate of 56 Kbps uses 7 bits to carry the payload information.

:SOURce[1..n]:DATA:TELEcom:PDH:E:MODE

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:E:ENAB ON* SOUR:DATA:TEL:PDH:E:MODE E64K* SOUR:DATA:TEL:PDH:E:MODE? Returns E64K
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled* SOURce[1..n]:DATA:TELEcom:PDH:E:MODE?

:SOURce[1..n]:DATA:TELEcom:PDH:E:MODE?

Description This query returns the channel timeslot data rate for the transmitter.

At *RST, this value is set to E64K.

Syntax :SOURce[1..n]:DATA:TELEcom:PDH:E:MODE?

Parameter(s) None

Response Syntax <Mode>

Response(s) Mode:
The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the channel timeslot data rate for the transmitter.
E64K, timeslot data rate of 64 Kbps uses 8 bits to carry the payload information is selected.
E56K, timeslot data rate of 56 Kbps uses 7 bits to carry the payload information is selected.

:SOURce[1..n]:DATA:TELEcom:PDH:E:MODE?

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:E:ENAB ON* SOUR:DATA:TEL:PDH:E:MODE E64K* SOUR:DATA:TEL:PDH:E:MODE? Returns E64K
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled* SOURce[1..n]:DATA:TELEcom:PDH:E:MODE

:SENSe[1..n]:DATA:TELEcom:PDH:E:MODE

Description This command selects the channel timeslot data rate for the receiver.

At *RST, this value is set to E64K.

Syntax :SENSe[1..n]:DATA:TELEcom:PDH:E:MODE
<wsp>E64K|E56K

Parameter(s) Mode:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: E64K|E56K.
Selects the channel timeslot data rate for the receiver.
E64K, selects a timeslot data rate of 64 Kbps uses 8 bits to carry the payload information.
E56K, selects a timeslot data rate of 56 Kbps uses 7 bits to carry the payload information.

:SENSe[1..n]:DATA:TELEcom:PDH:E:MODE

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:PDH:E:ENAB ON* SENS:DATA:TEL:PDH:E:MODE E64K* SENS:DATA:TEL:PDH:E:MODE? Returns E64K
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SENSE[1..n]:DATA:TELEcom:PDH:E:ENABLEd* SENSE[1..n]:DATA:TELEcom:PDH:E:MODE?

:SENSe[1..n]:DATA:TELEcom:PDH:E:MODE?

Description This query returns the channel timeslot data rate for the receiver.

At *RST, this value is set to E64K.

Syntax :SENSe[1..n]:DATA:TELEcom:PDH:E:MODE?

Parameter(s) None

Response Syntax <Mode>

Response(s) Mode:
The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the channel timeslot data rate for the receiver.
E64K, timeslot data rate of 64 Kbps uses 8 bits to carry the payload information is selected.
E56K, timeslot data rate of 56 Kbps uses 7 bits to carry the payload information is selected.

:SENSe[1..n]:DATA:TELecom:PDH:E:MODE?

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:PDH:E:ENAB ON* SENS:DATA:TEL:PDH:E:MODE E64K* SENS:DATA:TEL:PDH:E:MODE? Returns E64K
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELecom:PDH:E:ENABLEd* SENSe[1..n]:DATA:TELecom:PDH:E:MODE

:SOURce[1..n]:DATA:TELEcom:PDH:E:ZCS

Description This command sets Zero Code Suppression (ZCS) method used to replace the all-zero bytes of the idle and tone payload contents.

At *RST, this value is set to NONE.

Syntax :SOURce[1..n]:DATA:TELEcom:PDH:E:ZCS
<wsp>NONE|JBIT8

Parameter(s) Zcs:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NONE|JBIT8.
Sets the Zero Code Suppression (ZCS) method. NONE, No Zero Code Suppression method is selected.
JBIT8, selects every 8th (LSB) bit is forced to 1.

Example(s)

- * SOUR:DATA:TEL:PDH:E:ENAB ON
- * SOUR:DATA:TEL:PDH:E:ZCS NONE
- * SOUR:DATA:TEL:PDH:E:ZCS? Returns NONE

Note FTB/IQS-8140 Transport Blazer does not support this command.

See Also

- * SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled
- * SOURce[1..n]:DATA:TELEcom:PDH:E:ZCS?

:SOURce[1..n]:DATA:TELEcom:PDH:E:ZCS?

Description	<p>This query returns the Zero Code Suppression (ZCS) method used to replace the all-zero bytes of the idle and tone payload contents.</p> <p>At *RST, this value is set to NONE.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:PDH:E:ZCS?
Parameter(s)	None
Response Syntax	<Zcs>
Response(s)	<p>Zcs:</p> <p>The response data syntax for <Zcs> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Zero Code Suppression (ZCS) method.</p> <p>NONE, No Zero Code Suppression method is selected.</p> <p>JBIT8, Every 8th (LSB) bit is forced to 1 is selected.</p>

:SOURce[1..n]:DATA:TELEcom:PDH:E:ZCS?

Example(s)

- * SOUR:DATA:TEL:PDH:E:ENAB ON
- * SOUR:DATA:TEL:PDH:E:ZCS NONE
- * SOUR:DATA:TEL:PDH:E:ZCS? Returns NONE

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:PDH:E:ENABLEd
- * SOURce[1..n]:DATA:TELEcom:PDH:E:ZCS

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
IDLE**

Description	<p>This command sets the idle code byte from the idle field. The selected idle code applies to all timeslots set to idle.</p> <p>At *RST, this value is set to #H7F.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: IDLE<wsp><Idle></pre>
Parameter(s)	<p>Idle:</p> <p>The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the idle code byte from the idle field. The values are #H00 through #HFF.</p>
Example(s)	<pre>* SOUR:DATA:TEL:PDH:E:ENAB ON * SOUR:DATA:TEL:PDH:E:PAYL:IDLE #H3F * SOUR:DATA:TEL:PDH:E:PAYL:IDLE? Returns #H3F</pre>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this command.</p>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled * SOURce[1..n]:DATA:TELEcom:PDH:E: PAYLoad: IDLE?</pre>

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
IDLE?**

Description	<p>This query returns the idle code byte from the idle field.</p> <p>At *RST, this value is set to #H7F.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: IDLE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Idle></p>
Response(s)	<p>Idle: The response data syntax for <Idle> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element. Returns the idle code byte from the idle field. The values are #H00 to #HFF.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
IDLE?**

Example(s)	* SOUR:DATA:TEL:PDH:E:ENAB ON * SOUR:DATA:TEL:PDH:E:PAYL:IDLE #H3F * SOUR:DATA:TEL:PDH:E:PAYL:IDLE? Returns #H3F
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABLEd * SOURce[1..n]:DATA:TELEcom:PDH:E: PAYLoad: IDLE

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
TONE**

Description This command sets the tone for digital milli watt testing. The signal output power, when converted to analog, is 0 dBm. The selected tone applies to all timeslots set to tone.

At *RST, this value is set to T1004.

Syntax :SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
TONE<wsp>T1000|T1004

Parameter(s) Tone:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: T1000|T1004.
Sets the tone for digital milli watt testing.
T1000, selects the 1000 Hz tone.
T1004, selects the 1004 Hz tone.

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
TONE**

Example(s)	* SOUR:DATA:TEL:PDH:E:ENAB ON * SOUR:DATA:TEL:PDH:E:PAYL:TONE T1000 * SOUR:DATA:TEL:PDH:E:PAYL:TONE? Returns T1000
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled * SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:TONE?

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
TONE?**

Description	This query returns the tone for digital milli watt testing. At *RST, this value is set to T1004.
Syntax	:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: TONE?
Parameter(s)	None
Response Syntax	<Tone>
Response(s)	Tone: The response data syntax for <Tone> is defined as a <CHARACTER RESPONSE DATA> element. Returns the tone for digital milli watt testing. T1000, 1000 Hz tone is selected. T1004, 1004 Hz tone is selected.

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
TONE?**

Example(s)	* SOUR:DATA:TEL:PDH:E:ENAB ON * SOUR:DATA:TEL:PDH:E:PAYL:TONE T1000 * SOUR:DATA:TEL:PDH:E:PAYL:TONE? Returns T1000
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled * SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad :TONE

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent**

Description	<p>This command sets the payload content for transmitter.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent<wsp><Content> ,PATTERN1 IDLE TONE</p>
Parameter(s)	<p>Content: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the timeslots for payload content. Choices are 1 through 31.</p> <p>Payload: The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PATTERN1 IDLE TONE. Sets the payload type. PATTERN1, selects the Pattern as payload type. IDLE, selects the Idle as payload type. TONE, selects the Tone as payload type.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:E:ENAB ON* SOUR:DATA:TEL:PDH:E:PAYL:CONT 10,PAT1* SOUR:DATA:TEL:PDH:E:PAYL:CONT? 10 PATTERN1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled* SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad :CONTent?

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent?**

Description	<p>This query returns the payload content for transmitter.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent? <wsp> <Content></p>
Parameter(s)	<p>Content: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the timeslots for payload content. Choices are 1 through 31.</p>
Response Syntax	<p><Payload></p>
Response(s)	<p>Payload: The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element. Returns the payload type. PATTERN1, Pattern is selected as payload type. IDLE, Idle is selected as payload type. TONE, Tone is selected as payload type.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent?**

Example(s)	* SOUR:DATA:TEL:PDH:E:ENAB ON * SOUR:DATA:TEL:PDH:E:PAYL:CONT 10,PAT1 * SOUR:DATA:TEL:PDH:E:PAYL:CONT? 10 Returns PATTERN1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled * SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad :CONTent

:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent

Description	<p>This command sets the payload content for receiver.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent<wsp><Content> ,NONE PATTERN1</p>
Parameter(s)	<p>Content: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the timeslots for payload content. Choices are 1 through 31.</p> <p>Payload: The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NONE PATTERN1. Sets the payload content. NONE, No payload content. PATTERN1, selects the Pattern as payload content.</p>

**:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent**

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:PDH:E:ENAB ON* SENS:DATA:TEL:PDH:E:PAYL:CONT 10,NONE* SENS:DATA:TEL:PDH:E:PAYL:CONT? 10 Returns NONE
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:PDH:E:ENABled* SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad:CONTent?

**:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent?**

Description	<p>This query returns the payload content for receiver.</p> <p>At *RST, this value is set to PATTERN1.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent? <wsp> <Content></p>
Parameter(s)	<p>Content: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the timeslots for payload content. Choices are 1 through 31.</p>
Response Syntax	<p><Payload></p>
Response(s)	<p>Payload: The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element. Returns the payload. NONE, No payload content is selected. PATTERN1, Pattern is selected as payload content.</p>

**:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent?**

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:PDH:E:ENAB ON* SENS:DATA:TEL:PDH:E:PAYL:CONT 10,NONE* SENS:DATA:TEL:PDH:E:PAYL:CONT? 10 Returns NONE
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:PDH:E:ENABled* SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad:CONTent

:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE

Description

This command selects the type of payload content for transmitter.

At *RST, this value is set to PATTERN1.

Syntax

:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent:TYPE<wsp>PATTERN1|IDLE|TONE

Parameter(s)

Content:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

PATTERN1|IDLE|TONE.

Sets the payload content type.

PATTERN1, selects the Pattern as payload content type.

IDLE, selects the Idle as payload content type.

TONE, selects the Tone as payload content type.

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent:TYPE**

Example(s)	* SOUR:DATA:TEL:PDH:E:ENAB ON * SOUR:DATA:TEL:PDH:E:PAYL:CONT:TYPE PAT1 * SOUR:DATA:TEL:PDH:E:PAYL:CONT:TYPE? Returns PATTERN1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled * SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad :CONTent:TYPE?

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent:TYPE?**

Description	<p>This query returns the type of payload content for transmitter.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Content></p>
Response(s)	<p>Content: The response data syntax for the parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the payload content type. PATTERN1, Pattern is selected as payload content type. IDLE, Idle is selected as payload content type. TONE, Tone is selected as payload content type.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent:TYPE?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:E:ENAB ON* SOUR:DATA:TEL:PDH:E:PAYL:CONT:TYPE PAT1* SOUR:DATA:TEL:PDH:E:PAYL:CONT:TYPE? Returns PATTERN1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PDH:E:ENABled* SOURce[1..n]:DATA:TELEcom:PDH:E: PAYLoad:CONTent:TYPE* SOURce[1..n]:DATA:TELEcom:PDH:E: PAYLoad:CONTent:ALL

**:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent:TYPE**

Description	<p>This command selects the type of payload content for receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE<wsp>NONE PATTERN1</p>
Parameter(s)	<p>Content: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NONE PATTERN1. Sets the payload content type. NONE, No payload content type is selected. PATTERN1, selects the Pattern as payload content type.</p>

**:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent:TYPE**

Example(s)	* SENS:DATA:TEL:PDH:E:ENAB ON * SENS:DATA:TEL:PDH:E:PAYL:CONT:TYPE PAT1 * SENS:DATA:TEL:PDH:E:PAYL:CONT:TYPE? Returns PATTERN1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SENSe[1..n]:DATA:TELEcom:PDH:E:ENABled * SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE?

:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE?

Description	<p>This query returns the type of payload content for receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Content></p>
Response(s)	<p>Content: The response data syntax for the parameter is defined as a <CHARACTER RESPONSE DATA> element. Returns the payload content type. NONE, No payload content type is selected. PATTERN1, Pattern as payload content type is selected.</p>

**:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad:
CONTent:TYPE?**

Example(s)	* SENS:DATA:TEL:PDH:E:ENAB ON * SENS:DATA:TEL:PDH:E:PAYL:CONT:TYPE PAT1 * SENS:DATA:TEL:PDH:E:PAYL:CONT:TYPE? Returns PATTERN1
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SENSe[1..n]:DATA:TELEcom:PDH:E:ENABled * SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE * SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:ALL

:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:ALL

Description	<p>This command sets the payload content of all timeslots to the selected payload content with its Pattern, Idle, or Tone value.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:ALL</pre>
Parameter(s)	None
Example(s)	<pre>* SENS:DATA:TEL:PDH:E:ENAB ON * SOUR:DATA:TEL:PDH:E:PAYL:CONT:TYPE PAT1 * SOUR:DATA:TEL:PDH:E:PAYL:CONT:ALL</pre>
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad :CONTent:TYPE * SOURce[1..n]:DATA:TELEcom:PDH:E:PAYLoad :CONTent:TYPE?</pre>

:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:ALL

Description	<p>This command sets the payload content of all timeslots with (Pattern) or without (None) the selected Pattern.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:ALL
Parameter(s)	None
Example(s)	<ul style="list-style-type: none"> * SENS:DATA:TEL:PDH:E:ENAB ON * SENS:DATA:TEL:PDH:E:PAYL:CONT:TYPE PAT1 * SENS:DATA:TEL:PDH:E:PAYL:CONT:ALL
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none"> * SENSe[1..n]:DATA:TELEcom:PDH:E:ENABled * SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE * SENSe[1..n]:DATA:TELEcom:PDH:E:PAYLoad: CONTent:TYPE?

:SOURce[1..n]:DATA:TELEcom:PDH:E[1..n]: FRAMing

Description This command sets the E1/E2/E3/E4 framing that will be used for transmission.
(E - European standard for digital transmission-level)

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing<wsp>UNFRAMED1|PCM301|
PCM30C41|PCM311|PCM31C41|FRAMed

Parameter(s) Framing:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
UNFRAMED1|PCM301|PCM30C41|PCM311|
PCM31C41|FRAMed.
Sets the frame coding that will be used for transmission.
UNFRAMED1, selects the Unframed as E1/E2/E3/E4 framing.
PCM301, selects the PCM30 (Pulse Code Modulation) as E1/E2/E3/E4 framing.

**:SOURce[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing**

PCM30C41, selects the PCM30C41 as E1/E2/E3/E4 framing.

PCM311, selects the PCM311 as E1/E2/E3/E4 framing.

PCM31C41, selects the PCM31C4 1 as E1/E2/E3/E4 framing.

FRAMed, selects the Framed as E2/E3/E4 framing.

Example(s)

* SOUR:DATA:TEL:PDH:E1:FRAM UNF1

* SOUR:DATA:TEL:PDH:E1:FRAM?

Returns UNFRAMED1

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing?

**:SOURce[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing?**

Description	<p>This query returns the E1/E2/E3/E4 framing that will be used for transmission. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:E[1..n]: FRAMing?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Framing></p>
Response(s)	<p>Framing: The response data syntax for <Framing> is defined as a <CHARACTER RESPONSE DATA> element. Returns the frame coding that will be used for transmission. UNFRAMED1, Unframed is selected as E1/E2/E3/E4 framing. PCM301, Pulse Code Modulation (PCM) is selected as E1/E2/E3/E4 framing.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing?**

PCM30C41, PCM30C41 is selected as E1/E2/E3/E4 framing.

PCM311, PCM311 is selected as E1/E2/E3/E4 framing.

PCM31C41, PCM31C41 is selected as E1/E2/E3/E4 framing.

FRAMED, Framed is selected as E2/E3/E4 framing.

Example(s)

* SOUR:DATA:TEL:PDH:E1:FRAM UNF1

* SOUR:DATA:TEL:PDH:E1:FRAM?

Returns UNFRAMED1

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing

:SENSe[1..n]:DATA:TELEcom:PDH:E[1..n]: FRAMing

Description This command sets the E1/E2/E3/E4 framing that will be used for received signal.
(E - European standard for digital transmission-level)

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing <wsp> UNFRAMED1 | PCM301 |
PCM30C41 | PCM311 | PCM31C41 | FRAMed

Parameter(s) Framing:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
UNFRAMED1 | PCM301 | PCM30C41 | PCM311 | PCM31C41 | FRAMed.
Sets the frame coding that will be used for received signal.
UNFRAMED1, selects the Unframed as E1/E2/E3/E4 framing.
PCM301, selects the PCM (Pulse Code Modulation) as E1/E2/E3/E4 framing.
PCM30C41, selects the PCM30CRC41 as E1/E2/E3/E4 framing.

**:SENSe[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing**

PCM311, selects the PCM311 as E1/E2/E3/E4 framing.

PCM31C41, selects the PCM31C41 as E1/E2/E3/E4 framing.

FRAMed, selects the Framed as E2/E3/E4 framing.

Example(s)

* SENS:DATA:TEL:PDH:E1:FRAM UNF1

* SENS:DATA:TEL:PDH:E1:FRAM?

Returns UNFRAMED1

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SENSe[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing?

**:SENSE[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing?**

Description	<p>This query returns the E1/E2/E3/E4 framing that will be used for received signal. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSE[1..n]:DATA:TELEcom:PDH:E[1..n]: FRAMing?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Framing></p>
Response(s)	<p>Framing: The response data syntax for <Framing> is defined as a <CHARACTER RESPONSE DATA> element. Returns the frame coding that will be used for received signal. UNFRAMED1, Unframed is selected as E1/E2/E3/E4 framing. PCM301, Pulse Code Modulation (PCM) is selected as E1/E2/E3/E4 framing.</p>

**:SENSe[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing?**

PCM30C41, PCM30C41 is selected as E1/E2/E3/E4 framing.

PCM311, PCM311 is selected as E1/E2/E3/E4 framing.

PCM31C41, PCM31C41 is selected as E1/E2/E3/E4 framing.

FRAMED, Framed is selected as E2/E3/E4 framing.

Example(s)

* SENS:DATA:TEL:PDH:E1:FRAM UNF1

* SENS:DATA:TEL:PDH:E1:FRAM?

Returns UNFRAMED1

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SENSe[1..n]:DATA:TELEcom:PDH:E[1..n]:
FRAMing

**:SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:TYPE**

Description	<p>This command selects the type of E1/E2/E3/E4 alarm. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is set to AIS.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:TYPE<wsp>AIS RAI LOF3 RAIM LOMF TS16AIS CLOMf</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: AIS RAI LOF3 RAIM LOMF TS16AIS CLOMf.</p> <p>Selects the type of E1/E2/E3/E4 alarm.</p> <p>AIS, selects the AIS (Alarm Indication Signal) as E1/E2/E3/E4 alarm type.</p> <p>RAI, selects the RAI (Remote Alarm Indication) as E1/E2/E3/E4 alarm type.</p> <p>LOF3, selects the LOF (Loss of Frame) as E1/E2/E3/E4 alarm type.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:TYPE**

RAIM, selects the RAI-MF (Remote Alarm Indication Multiframe) as E1 alarm type.

LOMF, selects the LOMF (Loss of Multiframe) as E1 alarm type.

TS16AIS, selects the TS16 AIS (Alarm Indication Signal) as E1 alarm type.

CLOMf, selects the CRC (Cyclic Redundancy Check) LOMF (Loss of Multiframe) as E1 alarm type.

Example(s)

* SOUR:DATA:TEL:PDH:ALAR:E1:TYPE AIS

* SOUR:DATA:TEL:PDH:ALAR:E1:TYPE?

Returns AIS

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:TYPE?

**:SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:TYPE?**

Description	<p>This query returns the type of E1/E2/E3/E4 alarm. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is set to AIS.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:PDH:ALARm:E1:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of E1/E2/E3/E4 alarm.</p> <p>AIS, Alarm Indication Signal (AIS) is selected as E1/E2/E3/E4 alarm.</p> <p>RAI, Remote Alarm Indication (RAI) is selected as E1/E2/E3/E4 alarm.</p> <p>LOF3, Loss Of Frame (LOF) is selected as E1/E2/E3/E4 alarm.</p> <p>RAIM, Remote Alarm Indication Multiframe (RAIMF) is selected as E1 alarm.</p> <p>LOMF, Loss Of Multiframe (LOMF) is selected as E1 alarm.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:TYPE?**

TS16AIS, Timeslot 16 Alarm Indication Signal (TS16 AIS) is selected as E1 alarm.
CLOMf, Cyclic Redundancy Check Loss Of Multiframe (CRC LOMF) is selected as E1 alarm type.

Example(s)

* SOUR:DATA:TEL:PDH:ALAR:E1:TYPE AIS
* SOUR:DATA:TEL:PDH:ALAR:E1:TYPE?
Returns AIS

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:TYPE

**:SOURce[1..n]:DATA:TELEcom:PDH:ALARM:
E[1..n]**

Description This command enables or disables the status of E1/E2/E3/E4 alarm generation. (E - European standard for digital transmission-level)

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:PDH:ALARM:
E[1..n]<wsp><Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the E1 alarm generation.

**:SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:ALAR:E1:TYPE AIS* SOUR:DATA:TEL:PDH:ALAR:E1 ON* SOUR:DATA:TEL:PDH:ALAR:E1? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:TYPE* SOURce[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]?

:SOURce[1..n]:DATA:TELEcom:PDH:ALARM: E[1..n]?

Description	<p>This query returns the status of E1/E2/E3/E4 alarm generation. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:PDH:ALARM: E[1..n]?</code>
Parameter(s)	None
Response Syntax	<code><Set></code> Set: The response data syntax for the parameter is defined as a <code><NR1 NUMERIC RESPONSE DATA></code> element. Returns the status of E1 alarm generation.

**:SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]?****Example(s)**

- * SOUR:DATA:TEL:PDH:ALAR:E1:TYPE AIS
- * SOUR:DATA:TEL:PDH:ALAR:E1 ON
- * SOUR:DATA:TEL:PDH:ALAR:E1? Returns 1

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:TYPE
 - * SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]
-

**:SOURce[1..n]:DATA:TELEcom:PDH:ERROR:
E[1..n]:MANual:TYPE**

Description	<p>This command selects the manual type of E1/E2/E3/E4 error. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is set to FAS.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:ERROR: E[1..n]:MANual:TYPE<wsp>FAS CRC4 EBIT</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FAS CRC4 EBIT.</p> <p>Selects the type of E1/E2/E3/E4 error.</p> <p>FAS, selects the E1/E2/E3/E4 error as FAS (Frame Alignment Signal).</p> <p>CRC4, selects the E1 error as CRC4 [Cyclic Redundancy Check (a four-bit word that detects bit errors)].</p> <p>EBIT, selects the E1 error as EBIT (CRC-4 Error Signal).</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:MANual:TYPE**

Example(s)	* SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE FAS * SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE? Returns FAS
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:MANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:PDH:ERROR:
E[1..n]:MANual:TYPE?**

Description	<p>This query returns the manual type of E1/E2/E3/E4 error. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is set to FAS.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:ERROR: E[1..n]:MANual:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>
Response(s)	<p>Error: The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of E1/E2/E3/E4 error. FAS, Frame Alignment Signal (FAS) is selected as E1/E2/E3/E4 error. CRC4, Cyclical Redundancy Check (CRC 4) is selected as E1 error. EBIT, CRC-4 Error Signal (E-BIT) is selected as E1 error.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:MANual:TYPE?**

Example(s)	* SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE FAS * SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE? Returns FAS
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:MANual:TYPE

:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E[1..n]:AMOut

Description	<p>This command sets the amount of E1/E2/E3/E4 error to be injected into the instrument. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:AMOut<wsp><Amount> MAXimum MINimum</p>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the amount of E1/E2/E3/E4 error. Choices are 1 through 50.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AMOunt**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE FAS* SOUR:DATA:TEL:PDH:ERR:E1:AMO 15* SOUR:DATA:TEL:PDH:ERR:E1:AMO? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AMOunt?

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AMOUnt?**

Description	<p>This query returns the amount of E1/E2/E3/E4 error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1: AMOUnt? [<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<p><Amount></p>

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AMOunt?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of E1/E2/E3/E4 error.
Example(s)	* SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE FAS * SOUR:DATA:TEL:PDH:ERR:E1:AMO 15 * SOUR:DATA:TEL:PDH:ERR:E1:AMO? Returns 15
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AMOunt

:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:INJect

Description	<p>This command injects the type of E1/E2/E3/E4 error. (E - European standard for digital transmission-level)</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:INJect</code>
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE FAS* SOUR:DATA:TEL:PDH:ERR:E1:AMO 15* SOUR:DATA:TEL:PDH:ERR:E1:INJ
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AMOut

**:FETCh[1..n]:DATA:TELEcom:PDH:ERROr:
E[1..n]:HISTory?**

Description	<p>This query returns the history status of E1/E2/E3/E4 error. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PDH:ERROr:E[1..n]: HISTory? <wsp>FAS CRC4 EBIT</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FAS CRC4 EBIT.</p> <p>Selects the type of E1/E2/E3/E4 error.</p> <p>FAS, selects the FAS (Frame Alignment Signal) error which indicates that bits 2 to 8 of the frame containing the FAS differ from 0011011 for E1/E2/E3/E4.</p> <p>CRC4, selects the CRC-4 (Cyclical Redundancy Check) error which indicates that one or more bit errors are detected in a block of data through cyclical redundancy check for E1.</p>

**:FETCh[1..n]:DATA:TELEcom:PDH:ERROr:
E[1..n]:HISTOrY?**

EBIT, selects the E-Bit (CRC-4 Error Signal) error which indicates that bit 1 of SMF (sub-multiframe) II in frame 13 and/or 15 is set to 0 indicating a sub-multiframe error for E1.

Response Syntax <History>

Response(s)

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the history status of E1/E2/E3/E4 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.

**:FETCh[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:HISTory?**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE FAS* SOUR:DATA:TEL:PDH:ERR:E1:AMO 15* SOUR:DATA:TEL:PDH:ERR:E1:INJ* FETC:DATA:TEL:PDH:ERR:E1:HIST? FAS Returns the error history status.
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AMOUNT* SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:SEConds?**

Description This query returns the number of seconds within which E1/E2/E3/E4 error occurred.
(E - European standard for digital transmission-level)

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:PDH:ERRor:E[1..n]:
SEConds? <wsp> FAS | CRC4 | EBIT

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FAS|CRC4|EBIT.
Selects the type of E1/E2/E3/E4 error.
FAS, selects the FAS (Frame Alignment Signal) error which indicates that bits 2 to 8 of the frame containing the FAS differ from 0011011 for E1/E2/E3/E4.
CRC4, selects the CRC-4 (Cyclical Redundancy Check) error which indicates that one or more bit errors are detected in a block of data through cyclical redundancy check for E1.

**:FETCh[1..n]:DATA:TELEcom:PDH:ERROr:
E[1..n]:SEConds?**

EBIT, selects the E-Bit (CRC-4 Error Signal) error which indicates that bit 1 of SMF (sub-multiframe) II in frame 13 and/or 15 is set to 0 indicating a sub-multiframe error for E1.

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of errored seconds.

Example(s)

- * SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE FAS
- * SOUR:DATA:TEL:PDH:ERR:E1:AMO 15
- * SOUR:DATA:TEL:PDH:ERR:E1:INJ
- * FETC:DATA:TEL:PDH:ERR:E1:SEC? FAS

Returns the number of errored seconds.

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:PDH:ERROr:E[1..n]:MANual:TYPE
- * SOURce[1..n]:DATA:TELEcom:PDH:ERROr:E[1..n]:AMOUnt
- * SOURce[1..n]:DATA:TELEcom:PDH:ERROr:E[1..n]:INJect

:FETCh[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:CURRent?

Description	<p>This query returns the current status of E1/E2/E3/E4 error. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:PDH:ERRor:E[1..n]: CURRent? <wsp>FAS CRC4 EBIT</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FAS CRC4 EBIT.</p> <p>Selects the type of E1/E2/E3/E4 error.</p> <p>FAS, selects the FAS (Frame Alignment Signal) error which indicates that bits 2 to 8 of the frame containing the FAS differ from 0011011 for E1/E2/E3/E4.</p> <p>CRC4, selects the CRC-4 (Cyclical Redundancy Check) error which indicates that one or more bit errors are detected in a block of data through cyclical redundancy check for E1.</p>

**:FETCh[1..n]:DATA:TELEcom:PDH:ERror:
E[1..n]:CURrent?**

EBIT, selects the E-Bit (CRC-4 Error Signal) error which indicates that bit 1 of SMF (sub-multiframe) II in frame 13 and/or 15 is set to 0 indicating a sub-multiframe error for E1.

Response Syntax <Current>

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current status of E1/E2/E3/E4 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.

:FETCh[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:CURRent?

Example(s)

- * SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE FAS
- * SOUR:DATA:TEL:PDH:ERR:E1:AMO 15
- * SOUR:DATA:TEL:PDH:ERR:E1:INJ
- * FETC:DATA:TEL:PDH:ERR:E1:CURR? FAS

Returns the current error status.

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:MANual:TYPE
- * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AMOUNT
- * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:COUNT?**

Description	<p>This query returns the count of E1/E2/E3/E4 error. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PDH:ERRor:E[1..n]: COUNT?<wsp>FAS CRC4 EBIT</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FAS CRC4 EBIT.</p> <p>Selects the type of E1/E2/E3/E4 error.</p> <p>FAS, selects the FAS (Frame Alignment Signal) error which indicates that bits 2 to 8 of the frame containing the FAS differ from 0011011 for E1/E2/E3/E4.</p> <p>CRC4, selects the CRC-4 (Cyclical Redundancy Check) error which indicates that one or more bit errors are detected in a block of data through cyclical redundancy check for E1.</p>

:FETCh[1..n]:DATA:TELEcom:PDH:ERROR: E[1..n]:COUNT?

EBIT, selects the E-Bit (CRC-4 Error Signal) error which indicates that bit 1 of SMF (sub-multiframe) II in frame 13 and/or 15 is set to 0 indicating a sub-multiframe error for E1.

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of E1/E2/E3/E4 error.

Example(s)

- * SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE FAS
- * SOUR:DATA:TEL:PDH:ERR:E1:AMO 15
- * SOUR:DATA:TEL:PDH:ERR:E1:INJ
- * FETC:DATA:TEL:PDH:ERR:E1:COUN? FAS

Returns the error count.

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:PDH:ERROR:E[1..n]:MANual:TYPE
- * SOURce[1..n]:DATA:TELEcom:PDH:ERROR:E[1..n]:AMount
- * SOURce[1..n]:DATA:TELEcom:PDH:ERROR:E[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:RATE?**

Description	<p>This query returns the current rate of E1/E2/E3/E4 error. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PDH:ERRor:E[1..n]: RATE? <wsp>FAS CRC4 EBIT</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FAS CRC4 EBIT.</p> <p>Selects the type of E1/E2/E3/E4 error.</p> <p>FAS, selects the FAS (Frame Alignment Signal) error which indicates that bits 2 to 8 of the frame containing the FAS differ from 0011011 for E1/E2/E3/E4.</p> <p>CRC4, selects the CRC-4 (Cyclical Redundancy Check) error which indicates that one or more bit errors are detected in a block of data through cyclical redundancy check for E1.</p>

:FETCh[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:RATE?

EBIT, selects the E-Bit (CRC-4 Error Signal) error which indicates that bit 1 of SMF (sub-multiframe) II in frame 13 and/or 15 is set to 0 indicating a sub-multiframe error for E1.

Response Syntax <Rate>

Response(s) Rate:
The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.
Returns the current rate of E1/E2/E3/E4 error.

Example(s)

- * SOUR:DATA:TEL:PDH:ERR:E1:MAN:TYPE FAS
- * SOUR:DATA:TEL:PDH:ERR:E1:AMO 15
- * SOUR:DATA:TEL:PDH:ERR:E1:INJ
- * FETC:DATA:TEL:PDH:ERR:E1:RATE? FAS

Returns the error rate.

Note FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E[1..n]:MANual:TYPE
- * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E[1..n]:AMOUNT
- * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E[1..n]:INJect

:FETCh[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:HISTory?

Description	<p>This query returns the history status of E1/E2/E3/E4 alarm. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:HISTory?<wsp>LOF3 RAI AIS TS16AIS RAIM LOMF CLOMf</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF3 RAI AIS TS16AIS RAIM LOMF CLOMf.</p> <p>Selects the type of E1/E2/E3/E4 alarm.</p> <p>LOF3, selects the Loss of Frame (LOF) alarm to indicate that three consecutive incorrect frame alignment signals have been received for E1/E2/E3/E4.</p> <p>RAI, selects the Remote Alarm Indication (RAI) alarm when bit 3 in timeslot 0 is set to "1" for E1/E2/E3/E4.</p>

:FETCh[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:HISTory?

AIS, selects the Alarm Indication Signal (AIS) alarm when an unframed all-ones signal is received for E1/E2/E3/E4.

TS16AIS, selects the time slot 16 alarm indication signal alarm when timeslot 16 is received as all-ones for all frames of two consecutive multiframes for E1.

RAIM, selects the Remote Alarm Indication Multi-frame (RAI MF) alarm when bit 6 of timeslot 16 of frame 0 is set to "1".

LOMF, selects the Loss of Multi Frame (LOMF) alarm to indicate that two consecutive multiframes alignment signals (bits 1 through 4 of TS16 of frame 0) have been received with an error for E1.

CLOMf, selects the CRC LOMF (CRC Loss of Multi Frame) which indicates that the first bit of the NFAS in frames 1, 3, 5, 7, 9 and 11 differ from 0, 0, 1, 0, 1 and 1 respectively. CRC LOMF is available when the framing is set to PCM30 CRC-4 or PCM31 CRC-4 for E1.

Response Syntax <History>

**:FETCh[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:HISTory?****History:**

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the history status of E1/E2/E3/E4 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)

* SOUR:DATA:TEL:PDH:ALAR:E1:TYPE AIS

* SOUR:DATA:TEL:PDH:ALAR:E1 ON

* FETC:DATA:TEL:PDH:ALAR:E1:HIST? AIS

Returns the alarm history status.

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:TYPE

* SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]

**:FETCh[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:SEConds?**

Description

This query returns the number of seconds within which E1/E2/E3/E4 alarm occurred.
(E - European standard for digital transmission-level)

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:SEConds?<wsp>LOF3|RAI|AIS|TS16AIS
|RAIM|LOMF|CLOMf

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

LOF3|RAI|AIS|TS16AIS|RAIM|LOMF|CLOMf.

Selects the type of E1/E2/E3/E4 alarm.

LOF3, selects the Loss of Frame (LOF) alarm to indicate that three consecutive incorrect frame alignment signals have been received for E1/E2/E3/E4.

RAI, selects the Remote Alarm Indication (RAI) alarm when bit 3 in timeslot 0 is set to "1" for E1/E2/E3/E4.

**:FETCh[1..n]:DATA:TELeCom:PDH:ALARm:
E[1..n]:SEConds?**

AIS, selects the Alarm Indication Signal (AIS) alarm when an unframed all-ones signal is received for E1/E2/E3/E4.

TS16AIS, selects the time slot 16 alarm indication signal alarm when timeslot 16 is received as all-ones for all frames of two consecutive multiframes for E1.

RAIM, selects the Remote Alarm Indication Multi-frame (RAI MF) alarm when bit 6 of timeslot 16 of frame 0 is set to "1".

LOMF, selects the Loss of Multi Frame (LOMF) alarm to indicate that two consecutive multiframes alignment signals (bits 1 through 4 of TS16 of frame 0) have been received with an error for E1.

CLOMf, selects the CRC LOMF (CRC Loss of Multi Frame) which indicates that the first bit of the NFAS in frames 1, 3, 5, 7, 9 and 11 differ from 0, 0, 1, 0, 1 and 1 respectively. CRC LOMF is available when the framing is set to PCM30 CRC-4 or PCM31 CRC-4 for E1.

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of E1/E2/E3/E4 alarm.

:FETCh[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:SEConds?

Example(s)

- * SOUR:DATA:TEL:PDH:ALAR:E1:TYPE AIS
- * SOUR:DATA:TEL:PDH:ALAR:E1 ON
- * FETC:DATA:TEL:PDH:ALAR:E1:HIST? AIS

Returns the number of seconds of E1/E2/E3/E4 alarm.

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:TYPE
- * SOURce[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]

**:FETCh[1..n]:DATA:TELEcom:PDH:ALARm:
E[1..n]:CURRent?**

Description	<p>This query returns the current status of E1/E2/E3/E4 alarm. (E - European standard for digital transmission-level)</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:CURRent?<wsp>LOF3 RAI AIS TS16AIS RAIM LOMF CLOMf</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOF3 RAI AIS TS16AIS RAIM LOMF CLOMf.</p> <p>Selects the type of E1/E2/E3/E4 alarm.</p> <p>LOF3, selects the Loss of Frame (LOF) alarm to indicate that three consecutive incorrect frame alignment signals have been received for E1/E2/E3/E4.</p> <p>RAI, selects the Remote Alarm Indication (RAI) alarm when bit 3 in timeslot 0 is set to "1" for E1/E2/E3/E4.</p>

:FETCh[1..n]:DATA:TELEcom:PDH:ALARm: E[1..n]:CURRENT?

AIS, selects the Alarm Indication Signal (AIS) alarm when an unframed all-ones signal is received for E1/E2/E3/E4.

TS16AIS, selects the time slot 16 alarm indication signal alarm when timeslot 16 is received as all-ones for all frames of two consecutive multiframes for E1.

RAIM, selects the Remote Alarm Indication Multi-frame (RAI MF) alarm when bit 6 of timeslot 16 of frame 0 is set to "1".

LOMF, selects the Loss of Multi Frame (LOMF) alarm to indicate that two consecutive multiframes alignment signals (bits 1 through 4 of TS16 of frame 0) have been received with an error for E1.

CLOMf, selects the CRC LOMF (CRC Loss of Multi Frame) which indicates that the first bit of the NFAS in frames 1, 3, 5, 7, 9 and 11 differ from 0, 0, 1, 0, 1 and 1 respectively. CRC LOMF is available when the framing is set to PCM30 CRC-4 or PCM31 CRC-4 for E1.

Response Syntax <Current>

**:FETCh[1..n]:DATA:TELeom:PDH:ALARm:
E[1..n]:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of E1/E2/E3/E4 alarm. 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:ALAR:E1:TYPE AIS* SOUR:DATA:TEL:PDH:ALAR:E1 ON* FETC:DATA:TEL:PDH:ALAR:E1:CURR? AIS <p>Returns the current alarm status.</p>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELeom:PDH:ALARm:E[1..n]:TYPE* SOURce[1..n]:DATA:TELeom:PDH:ALARm:E[1..n]

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AUTomated:TYPE**

Description	<p>This command selects the type of E1/E2/E3/E4 error for automated injection.</p> <p>At *RST, this value is set to FAS.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:TYPE<wsp>FAS CRC4 EBIT</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FAS CRC4 EBIT.</p> <p>Selects the type of E1/E2/E3/E4 error for automated injection.</p> <p>FAS, selects the E1/E2/E3/E4 error as FAS (Frame Alignment Signal).</p> <p>CRC4, selects the E1 error as CRC4 [Cyclic Redundancy Check (a four-bit word that detects bit errors)].</p> <p>EBIT, selects the E1 error as EBIT (CRC-4 Error Signal).</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:
AUTomated:TYPE**

Example(s)	* SOUR:DATA:TEL:PDH:ERR:E1:AUT:TYPE FAS * SOUR:DATA:TEL:PDH:ERR:E1:AUT:TYPE? Returns FAS
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1: AUTomated:TYPE? * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1: AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1: AUTomated

:SOURce[1..n]:DATA:TELecom:PDH:ERROr: E[1..n]:AUTomated:TYPE?

Description	<p>This query returns the type of E1/E2/E3/E4 error for automated injection.</p> <p>At *RST, this value is set to FAS.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELecom:PDH:ERROr: E[1..n]:AUTomated:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>
Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the type of E1/E2/E3/E4 error for the automated injection.</p> <p>FAS, Frame Alignment Signal (FAS) is selected as E1/E2/E3/E4 error.</p> <p>CRC4, Cyclical Redundancy Check (CRC 4) is selected as E1 error.</p> <p>EBIT, Error Signal (E-BIT) is selected as E1 error.</p>

:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:AUTomated:TYPE?

Example(s)	* SOUR:DATA:TEL:PDH:ERR:E1:AUT:TYPE FAS * SOUR:DATA:TEL:PDH:ERR:E1:AUT:TYPE? Returns FAS
Note	FTB/IQS-8140 Transport Blazer does not support this query.
See Also	* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:AUTomated

:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:RATE

Description	<p>This command sets the injection rate for the selected FE1/E2/E3/E4 error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:RATE <wsp> <Rate> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected E1/E2/E3/E4 error.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AUTomated:RATE**

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:ERR:E1:AUT:TYPE FAS* SOUR:DATA:TEL:PDH:ERR:E1:AUT:RATE 1.0E-09* SOUR:DATA:TEL:PDH:ERR:E1:AUT:RATE? Returns 1.0E-09
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1: AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1: AUTomated:RATE?* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1: AUTomated

:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:RATE?

Description	<p>This query returns the injection rate for the selected E1/E2/E3/E4 error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:RATE?[<wsp>MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injection rate will be returned.</p>
Response Syntax	<p><Rate></p>

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AUTomated:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected E1/E2/E3/E4 error.</p>
Example(s)	<pre>* SOUR:DATA:TEL:PDH:ERR:E1:AUT:TYPE FAS * SOUR:DATA:TEL:PDH:ERR:E1:AUT:RATE 1.0E-09 * SOUR:DATA:TEL:PDH:ERR:E1:AUT:RATE? Returns 1.0E-09</pre>
Note	<p>FTB/IQS-8140 Transport Blazer does not support this query.</p>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1: AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1: AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1: AUTomated</pre>

**:SOURce[1..n]:DATA:TELEcom:PDH:ERROR:
E[1..n]:AUTomated**

Description This command enables or disables the selected automated E1/E2/E3/E4 error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:PDH:ERROR:
E[1..n]:AUTomated<wsp><Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the automated E1/E2/E3/E4 error injection.

:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E[1..n]:AUTomated

Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:PDH:ERR:E1:AUT:TYPE FAS* SOUR:DATA:TEL:PDH:ERR:E1:AUT:RATE 1.0E-09* SOUR:DATA:TEL:PDH:ERR:E1:AUT ON* SOUR:DATA:TEL:PDH:ERR:E1:AUT? Returns 1
Note	FTB/IQS-8140 Transport Blazer does not support this command.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AUTomated?**

Description	This query returns the status of automated E1/E2/E3/E4 error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated E1/E2/E3/E4 error injection.

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AUTomated?****Example(s)**

- * SOUR:DATA:TEL:PDH:ERR:E1:AUT:TYPE FAS
- * SOUR:DATA:TEL:PDH:ERR:E1:AUT:RATE
1.0E-09
- * SOUR:DATA:TEL:PDH:ERR:E1:AUT ON
- * SOUR:DATA:TEL:PDH:ERR:E1:AUT? Returns 1

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

- * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:
AUTomated:TYPE
 - * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:
AUTomated:RATE
 - * SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:
AUTomated
-

:SOURce[1..n]:DATA:TELEcom:PDH:ERRor: E[1..n]:AUTomated:CONTInuous

Description This command enables or disables the continuous rate of automated E1/E2/E3/E4 error injection.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AUTomated:CONTInuous<wsp><Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the rate of automated E1/E2/E3/E4 error injection continuously.

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AUTomated:CONTInuous****Example(s)**

* SOUR:DATA:TEL:PDH:ERR:E1:AUT:TYPE FAS
* SOUR:DATA:TEL:PDH:ERR:E1:AUT:CONT ON
* SOUR:DATA:TEL:PDH:ERR:E1:AUT:CONT?
Returns 1
* SOUR:DATA:TEL:PDH:ERR:E1:AUT ON

Note

FTB/IQS-8140 Transport Blazer does not support this command.

See Also

* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:
AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:
AUTomated
* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:
AUTomated:CONTInuous?

:SOURce[1..n]:DATA:TELEcom:PDH:ERROR: E[1..n]:AUTomated:CONTInuous?

Description	<p>This query returns the status of continuous rate of automated E1/E2/E3/E4 error injection.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:PDH:ERROR: E[1..n]:AUTomated:CONTInuous?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of continuous rate of automated E1/E2/E3/E4 error injection.</p>

**:SOURce[1..n]:DATA:TELEcom:PDH:ERRor:
E[1..n]:AUTomated:CONTInuous?****Example(s)**

* SOUR:DATA:TEL:PDH:ERR:E1:AUT:TYPE FAS
* SOUR:DATA:TEL:PDH:ERR:E1:AUT:CONT ON
* SOUR:DATA:TEL:PDH:ERR:E1:AUT:CONT?
Returns 1
* SOUR:DATA:TEL:PDH:ERR:E1:AUT ON

Note

FTB/IQS-8140 Transport Blazer does not support this query.

See Also

* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:
AUTomated:TYPE
* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:
AUTomated
* SOURce[1..n]:DATA:TELEcom:PDH:ERRor:E1:
AUTomated:CONTInuous

:FETCh[1..n]:DATA:TELEcom:PDH:E[1..n]:PM:STATistics?

Description

This query returns the performance monitoring statistics of European standard for digital transmission levels.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:PDH:E[1..n]:PM:
STATistics?<wsp>G826ISM|M2100ISM,EFs|EB|
ES|SES|BBE|UAS|ESR|SESR|BBER,NEND|
FEND
```

Parameter(s)

Standard:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
G826ISM|M2100ISM.

Selects the performance monitoring standard number.

G826ISM, selects G.826 ISM as a standard number.

M2100ISM, selects M.2100 ISM as a standard number.

**:FETCh[1..n]:DATA:TELecom:PDH:E[1..n]:PM:
STATistics?**

Type:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

EFS|EB|ES|SES|BBE|UAS|ESR|SESR|BBER.

Selects the performance monitoring statistics.

EFS, selects Error Free Seconds (EFS).

EB, selects Errored Block (EB).

ES, selects Errored Seconds (ES).

SES, selects Severely Errored Seconds (SES).

BBE, selects Background Block Error (SES).

UAS, selects Unavailable Second (UAS).

ESR, selects Errored Second Ratio (ESR).

SESR, selects Severely Errored Second Ratio (SESR).

BBER, selects Background Block Error Ratio (BBER).

**:FETCh[1..n]:DATA:TELEcom:PDH:E[1..n]:PM:
STATistics?**

End:

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

NEND|FEND.

Selects Near-End or Far-End.

NEND, selects the standard for Near-End.

FEND, selects the standard for Far-End.

Response Syntax <Statistics>

Response(s) Statistics:

The response data syntax for <Statistics> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the performance monitoring statistics of European standard for digital transmission levels.

Example(s) * FETC:DATA:TEL:PDH:E1:PM:STAT?
G826ISM, EFS, NEND

Returns the performance monitoring statistics of European standard for digital transmission level 1.

SDT and RTD Command Reference

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

Description	<p>This command selects on which layer the service disruption time test will be performed for SONET.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDT:SONet:LAYer: TYPE<wsp>PORT SECTion LINE HOP LOP DS1 DS3 E1 E2 E3 E4 PATtern</pre>
Parameter(s)	<p>Layer:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PORT SECTion LINE HOP LOP DS1 DS3 E1 E2 E3 E4 PATtern.</p> <p>Selects on which layer the service disruption time test will be performed. Choices depend on the selected test path.</p> <p>PORT, selects the layer on which the service disruption time test will be performed as Port.</p> <p>SECTion, selects the layer on which the service disruption time test will be performed as Section.</p>

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

LINE, selects the layer on which the service disruption time test will be performed as Line.
HOP, selects the layer on which the service disruption time test will be performed as HOP.
LOP, selects the layer on which the service disruption time test will be performed as LOP.
DS1, selects the layer on which the service disruption time test will be performed as DS1.
DS3, selects the layer on which the service disruption time test will be performed as DS3.
E1, selects the layer on which the service disruption time test will be performed as E1.
E2, selects the layer on which the service disruption time test will be performed as E2.
E3, selects the layer on which the service disruption time test will be performed as E3.
E4, selects the layer on which the service disruption time test will be performed as E4.
PATTERn, selects the layer on which the service disruption time test will be performed as Pattern.

Example(s)

* SENS:DATA:TEL:SDT:SON:LAY:TYPE SECT
* SENS:DATA:TEL:SDT:SON:LAY:TYPE?
Returns SECTION

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

See Also

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

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

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

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

Description	<p>This query returns the layer for service disruption time test for SONET.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SDT:SONet:LAYer: TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Layer></p>
Response(s)	<p>Layer:</p> <p>The response data syntax for <Layer> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the layer for service disruption time test will be performed.</p> <p>PORT, selects the layer on which the service disruption time test will be performed as Port.</p> <p>SECTION, selects the layer on which the service disruption time test will be performed as Section.</p> <p>LINE, selects the layer on which the service disruption time test will be performed as Line.</p>

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

HOP, selects the layer on which the service disruption time test will be performed as HOP.

LOP, selects the layer on which the service disruption time test will be performed as LOP.

DS1, selects the layer on which the service disruption time test will be performed as DS1.

DS3, selects the layer on which the service disruption time test will be performed as DS3.

E1, selects the layer on which the service disruption time test will be performed as E1.

E2, selects the layer on which the service disruption time test will be performed as E2.

E3, selects the layer on which the service disruption time test will be performed as E3.

E4, selects the layer on which the service disruption time test will be performed as E4.

PATTERN, selects the layer on which the service disruption time test will be performed as Pattern.

Example(s)

* SENS:DATA:TEL:SDT:SON:LAY:TYPE SECT

* SENS:DATA:TEL:SDT:SON:LAY:TYPE?

Returns SECTION

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

See Also

* SENSE[1..n]:DATA:TELEcom:SDT:SONet:LAYer:
TYPE

* SENSE[1..n]:DATA:TELEcom:SDT:SONet:
DSElection

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

:SENSe[1..n]:DATA:TELEcom:SDT:SONet: DSElection

Description

This command selects the defect of SONET for specific layer.

At *RST, this value is device dependent.

Syntax

```
:SENSe[1..n]:DATA:TELEcom:SDT:SONet:
DSElection<wsp>LOS|LOF|SEF|B1|AISL|RDI
L|B2|REIL|AISP|RDIP|EPSD|EPCD|EPPD|LOM
|LOPP|PDIP|B3|REIP|AISV|RDIV|EVSD|EVCD|
EVPD|RFIV|LOPV|BIP2|REIV|E1AIS|E1RAI|
E1LOF|RAIMf|LOMF|TS16AIS|E1FAS|E2AIS|
E2RAI|E2LOF|E2FAS|E3AIS|E3RAI|E3LOF|
E3FAS|E4AIS|E4RAI|E4LOF|E4FAS|DS1AIS|
RAI|DS1OOF|FRAMingbit|CRC6|DS3AIS|RDI|
DS3OOF|IDLE|CBIT|FBIT|PBIT|FEBE|PLOSs|
BITerror|OPUCSF|OPUAIS
```

**:SENSe[1..n]:DATA:TELecom:SDT:SONet:
DSElection****Parameter(s)**

Selection:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

LOS|LOF|SEF|B1|AISL|RDIL|B2|REIL|AISP|
RDIP|EPSD|EPCD|EPPD|LOM|LOPP|PDIP|B3|
REIP|AISV|RDIV|EVSD|EVCD|EVPD|RFIV|
LOPV|BIP2|REIV|E1AIS|E1RAI|E1LOF|RAIMf|
LOMF|TS16AIS|E1FAS|E2AIS|E2RAI|E2LOF|
E2FAS|E3AIS|E3RAI|E3LOF|E3FAS|E4AIS|
E4RAI|E4LOF|E4FAS|DS1AIS|RAI|DS1OOF|
FRAMingbit|CRC6|DS3AIS|RDI|DS3OOF|IDLE|
CBIT|FBIT|PBIT|FEBE|PLOSs|BITerror|
OPUCSF|OPUAIS

Selects the defect for SONET layer.

LOS, selects the defect for SONET layer as LOS.

LOF, selects the defect for SONET layer as LOF.

SEF, selects the defect for SONET layer as SEF.

B1, selects the defect for SONET layer as B1.

AISL, selects the defect for SONET layer as AIS-L.

RDIL, selects the defect for SONET layer as RDI-L.

B2, selects the defect for SONET layer as B2.

REIL, selects the defect for SONET layer as REI-L.

AISP, selects the defect for SONET layer as AIS-P.

RDIP, selects the defect for SONET layer as RDI-P.

EPSD, selects the defect for SONET layer as ERDI-PSD.

**:SENSe[1..n]:DATA:TELecom:SDT:SONet:
DSElection**

EPCD, selects the defect for SONET layer as ERDI-PCD.

EPPD, selects the defect for SONET layer as ERDI-PPD.

LOM, selects the defect for SONET layer as LOM.

LOPP, selects the defect for SONET layer as LOPP.

PDIP, selects the defect for SONET layer as PDIP.

B3, selects the defect for SONET layer as B3.

REIP, selects the defect for SONET layer as REI-P.

AISV, selects the defect for SONET layer as AIS-V.

RDIV, selects the defect for SONET layer as RDI-V.

EVSD, selects the defect for SONET layer as ERDI-VCD.

EVCD, selects the defect for SONET layer as ERDI-VCD.

EVPD, selects the defect for SONET layer as ERDI-VCD.

RFIV, selects the defect for SONET layer as RFIV.

LOPV, selects the defect for SONET layer as LOPV.

BIP2, selects the defect for SONET layer as BIP2.

REIV, selects the defect for SONET layer as REI-V.

E1AIS, selects the defect for SONET layer as AIS.

E1RAI, selects the defect for SONET layer as RAI.

**:SENSe[1..n]:DATA:TELecom:SDT:SONet:
DSELection**

E1LOF, selects the defect for SONET layer as LOF.

RAIMf, selects the defect for SONET layer as RAIMF.

LOMF, selects the defect for SONET layer as LOMF.

OPUCSF, selects the defect for SONET layer as OPUCSF.

OPUAIS, selects the defect for SONET layer as OPUAIS.

Example(s)

* SENS:DATA:TEL:SDT:SON:LAY:TYPE SECT

* SENS:DATA:TEL:SDT:SON:DSEL BERR

* SENS:DATA:TEL:SDT:SON:DSEL?

Returns BERROR

See Also

* SENSe[1..n]:DATA:TELecom:SDT:SONet:LAYer:
TYPE

* SENSe[1..n]:DATA:TELecom:SDT:SONet:
DSELection?

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

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

Description	This query returns the defect of SONET for specific layer. At *RST, this value is device dependent.
Syntax	:SENSe[1..n]:DATA:TELEcom:SDT:SONet: DSElection?
Parameter(s)	None
Response Syntax	<Selection>

:SENSe[1..n]:DATA:TELecom:SDT:SONet: DSElection?

Response(s)

Selection:

The response data syntax for <Selection> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the defect for SONET layer.

LOS, selects the defect for SONET layer as LOS.

LOF, selects the defect for SONET layer as LOF.

SEF, selects the defect for SONET layer as SEF.

B1, selects the defect for SONET layer as B1.

AISL, selects the defect for SONET layer as AIS-L.

RDIL, selects the defect for SONET layer as RDI-L.

B2, selects the defect for SONET layer as B2.

REIL, selects the defect for SONET layer as REI-L.

AISP, selects the defect for SONET layer as AIS-P.

RDIP, selects the defect for SONET layer as RDI-P.

EPSP, selects the defect for SONET layer as ERDI-PSD .

EPCD, selects the defect for SONET layer as ERDI-PCD.

EPPD, selects the defect for SONET layer as ERDI-PPD.

LOM, selects the defect for SONET layer as LOM.

LOPP, selects the defect for SONET layer as LOPP.

PDIP, selects the defect for SONET layer as PDIP.

B3, selects the defect for SONET layer as B3.

REIP, selects the defect for SONET layer as REI-P.

AISV, selects the defect for SONET layer as AIS-V.

RDIV, selects the defect for SONET layer as RDI-V.

EVSD, selects the defect for SONET layer as ERDI-VCD.

:SENSe[1..n]:DATA:TELecom:SDT:SONet: DSELection?

EVCD, selects the defect for SONET layer as ERDI-VCD.

EVPD, selects the defect for SONET layer as ERDI-VCD.

RFIV, selects the defect for SONET layer as RFIV.

LOPV, selects the defect for SONET layer as LOPV.

BIP2, selects the defect for SONET layer as BIP2.

REIV, selects the defect for SONET layer as REI-V.

E1AIS, selects the defect for SONET layer as AIS.

E1RAI, selects the defect for SONET layer as RAI.

E1LOF, selects the defect for SONET layer as LOF.

RAIMF, selects the defect for SONET layer as RAIMF.

LOMF, selects the defect for SONET layer as LOMF.

OPUCSF, selects the defect for SONET layer as OPUCSF.

OPUAIS, selects the defect for SONET layer as OPUAIS.

Example(s)

* SENS:DATA:TEL:SDT:SON:LAY:TYPE SECT

* SENS:DATA:TEL:SDT:SON:DSEL BERR

* SENS:DATA:TEL:SDT:SON:DSEL?

Returns BERROR

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

See Also

- * SENSe[1..n]:DATA:TELEcom:SDT:SONet:LAYer:
TYPE
 - * SENSe[1..n]:DATA:TELEcom:SDT:SONet:
DSElection
 - * SENSe[1..n]:DATA:TELEcom:SDT
-

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

Description	<p>This command selects on which layer the service disruption time test will be performed for SDH.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDT:SDH:LAYer: TYPE<wsp>PORT RS MS HOP LOP DS1 DS3 E1 E2 E3 E4 PATtern</pre>
Parameter(s)	<p>Layer:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PORT RS MS HOP LOP DS1 DS3 E1 E2 E3 E4 PATtern.</p> <p>Selects on which layer the service disruption time test will be performed. Choices depend on the selected test path.</p> <p>PORT, selects the layer on which the service disruption time test will be performed as Port.</p> <p>RS, selects the layer on which the service disruption time test will be performed as RS.</p> <p>MS, selects the layer on which the service disruption time test will be performed as MS.</p> <p>HOP, selects the layer on which the service disruption time test will be performed as HOP.</p>

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

LOP, selects the layer on which the service disruption time test will be performed as LOP.
DS1, selects the layer on which the service disruption time test will be performed as DS1.
DS3, selects the layer on which the service disruption time test will be performed as DS3.
E1, selects the layer on which the service disruption time test will be performed as E1.
E2, selects the layer on which the service disruption time test will be performed as E2.
E3, selects the layer on which the service disruption time test will be performed as E3.
E4, selects the layer on which the service disruption time test will be performed as E4.
PATtern, selects the layer on which the service disruption time test will be performed as Pattern.

Example(s)

* SENS:DATA:TEL:SDT:SDH:LAY:TYPE PORT
* SENS:DATA:TEL:SDT:SDH:LAY:TYPE
Returns PORT

See Also

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

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

Description	<p>This query returns the layer for service disruption time test for SDH.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:SDT:SDH:LAYer: TYPE?
Parameter(s)	None
Response Syntax	<Layer>
Response(s)	<p>Layer:</p> <p>The response data syntax for <Layer> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the layer for service disruption time test will be performed.</p> <p>PORT, selects the layer on which the service disruption time test will be performed as Port.</p> <p>RS, selects the layer on which the service disruption time test will be performed as RS.</p> <p>MS, selects the layer on which the service disruption time test will be performed as MS.</p> <p>HOP, selects the layer on which the service disruption time test will be performed as HOP.</p>

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

LOP, selects the layer on which the service disruption time test will be performed as LOP.
DS1, selects the layer on which the service disruption time test will be performed as DS1.
DS3, selects the layer on which the service disruption time test will be performed as DS3.
E1, selects the layer on which the service disruption time test will be performed as E1.
E2, selects the layer on which the service disruption time test will be performed as E2.
E3, selects the layer on which the service disruption time test will be performed as E3.
E4, selects the layer on which the service disruption time test will be performed as E4.
PATTERN, selects the layer on which the service disruption time test will be performed as Pattern.

Example(s)

* SENS:DATA:TEL:SDT:SDH:LAY:TYPE PORT
* SENS:DATA:TEL:SDT:SDH:LAY:TYPE
Returns PORT

See Also

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

:SENSe[1..n]:DATA:TELEcom:SDT:SDH: DSElection

Description

This command selects the defect for specific layer for SDH.

At *RST, this value is device dependent.

Syntax

```
:SENSe[1..n]:DATA:TELEcom:SDT:SDH:
DSElection<wsp>LOS|BPV|LOF|OOF|B1|
MSAis|MSRDi|B2|MSREi|AUAis|AULop|H4LOM
|HPRDi|ESD|ECD|EPD|B3|HPRi|TUAis|
TULop|LPRFi|LPRDi|ESDLop|ECDLop|EPDLop
|BIP2|LPREi|E1AIS|E1RAI|E1LOF|RAIMf|LOMF
|TS16AIS|E1FAS|E2AIS|E2RAI|E2LOF|E2FAS|
E3AIS|E3RAI|E3LOF|E3FAS|E4AIS|E4RAI|
E4LOF|E4FAS|DS1AIS|RAI|DS1OOF|
FRAMingbit|CRC6|DS3AIS|RDI|DS3OOF|IDLE|
CBIT|FBIT|PBIT|FEBE|PLOSs|BITerror
```

**:SENSE[1..n]:DATA:TELEcom:SDT:SDH:
DSElection****Parameter(s)****Selection:**

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

LOS|BPV|LOF|OOF|B1|
MSAis|MSRDi|B2|MSREi|AUAis|AULop|H4LOM
|HPRDi|ESD|ECD|EPD|B3|HPRi|TUAis|
TULop|LPRFi|LPRDi|ESDLop|ECDLop|EPDLop
|BIP2|LPREi|E1AIS|E1RAI|E1LOF|RAIMf|LOMF
|TS16AIS|E1FAS|E2AIS|E2RAI|E2LOF|E2FAS|
E3AIS|E3RAI|E3LOF|E3FAS|E4AIS|E4RAI|
E4LOF|E4FAS|DS1AIS|RAI|DS1OOF|
FRAMingbit|CRC6|DS3AIS|RDI|DS3OOF|IDLE|
CBIT|FBIT|PBIT|FEBE|PLOSs|BITerror.

Selects the defect for SDH layer.

LOS, selects the defect for SDH layer as LOS.

BPV, selects the defect for SDH layer as BPV.

LOF, selects the defect for SDH layer as LOF.

OOF, selects the defect for SDH layer as OOF.

B1, selects the defect for SDH layer as B1.

MSAis, selects the defect for SDH layer as MSAis.

MSRDi, selects the defect for SDH layer as MSRDi.

B2, selects the defect for SDH layer as B2.

MSREi, selects the defect for SDH layer as MSREi.

AUAis, selects the defect for SDH layer as AUAis.

AULop, selects the defect for SDH layer as AULop.

H4LOM, selects the defect for SDH layer as H4LOM.

HPRDi, selects the defect for SDH layer as HPRDi.

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

ESD, selects the defect for SDH layer as ESD.
ECD, selects the defect for SDH layer as ECD.
EPD, selects the defect for SDH layer as EPD.
B3, selects the defect for SDH layer as B3.
HPRei, selects the defect for SDH layer as HPRei.
TUAis, selects the defect for SDH layer as TUAis.
TULop, selects the defect for SDH layer as TULop.
LPRFi, selects the defect for SDH layer as LPRFi.
LPRDi, selects the defect for SDH layer as LPRDi.
ESDLop, selects the defect for SDH layer as ESDLop.
ECDLop, selects the defect for SDH layer as ECDLop.
EPDLop, selects the defect for SDH layer as EPDLop.
BIP2, selects the defect for SDH layer as BIP2.
LPREi, selects the defect for SDH layer as LPREi.
E1AIS, selects the defect for SDH layer as E1AIS.
E1RAI, selects the defect for SDH layer as E1RAI.
E1LOF, selects the defect for SDH layer as E1LOF.
RAIMf, selects the defect for SDH layer as RAIMf.
LOMF, selects the defect for SDH layer as LOMF.
TS16AIS, selects the defect for SDH layer as TS16AIS.
E1FAS, selects the defect for SDH layer as E1FAS.
E2AIS, selects the defect for SDH layer as E2AIS.
E2RAI, selects the defect for SDH layer as E2RAI.
E2LOF, selects the defect for SDH layer as E2LOF.
E2FAS, selects the defect for SDH layer as E2FAS.
E3AIS, selects the defect for SDH layer as E3AIS.

:SENSE[1..n]:DATA:TELEcom:SDT:SDH: DSElection

E3RAI, selects the defect for SDH layer as E3RAI.
E3LOF, selects the defect for SDH layer as E3LOF.
E3FAS, selects the defect for SDH layer as E3FAS.
E4AIS, selects the defect for SDH layer as E4AIS.
E4RAI, selects the defect for SDH layer as E4RAI.
E4LOF, selects the defect for SDH layer as E4LOF.
E4FAS, selects the defect for SDH layer as
DS1AIS.
DS1AIS, selects the defect for SDH layer as
DS1AIS.
RAI, selects the defect for SDH layer as RAI.
DS1OOF, selects the defect for SDH layer as
DS1OOF.
FRAMingbit, selects the defect for SDH layer as
FRAMingbit.
CRC6, selects the defect for SDH layer as CRC6.
DS3AIS, selects the defect for SDH layer as
DS3AIS.
RDI, selects the defect for SDH layer as RDI.
DS3OOF, selects the defect for SDH layer as
DS3OOF.
IDLE, selects the defect for SDH layer as IDLE.
CBIT, selects the defect for SDH layer as CBIT.
FBIT, selects the defect for SDH layer as FBIT.
PBIT, selects the defect for SDH layer as PBIT.
FEBE, selects the defect for SDH layer as FEBE.
PLOSs, selects the defect for SDH layer as Pattern
Loss.
BITerror, selects the defect for SDH layer as Bit
error.

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

Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:SDH:LAY:TYPE PORT* SENS:DATA:TEL:SDT:SDH:DSEL LOS* SENS:DATA:TEL:SDT:SDH:DSEL? Returns LOS
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:SDT:SDH:LAYer:TYPE* SENSe[1..n]:DATA:TELEcom:SDT:SDH:DSElection?* SENSe[1..n]:DATA:TELEcom:SDT7

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

Description	<p>This query returns the defect of SDH for specific layer.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SDT:SDH: DSElection?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Selection></p>
Response(s)	<p>Selection: The response data syntax for <Selection> is defined as a <CHARACTER RESPONSE DATA> element. Returns the defect for SDH layer. LOS, LOS is selected as defect for SDH layer. BPV, BPV is selected as defect for SDH layer. LOF, LOF is selected as defect for SDH layer. OOF, OOF is selected as defect for SDH layer. B1, B1 is selected as defect for SDH layer. MSAis, MSAis is selected as defect for SDH layer. MSRDi, MSRDi is selected as defect for SDH layer. B2, B2 is selected as defect for SDH layer. MSREi, MSREi is selected as defect for SDH layer. AUAis, AUAis is selected as defect for SDH layer.</p>

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

AULop, AULop is selected as defect for SDH layer.

H4LOM, H4LOM is selected as defect for SDH layer.

HPRDi, HPRDi is selected as defect for SDH layer.

ESD, ESD is selected as defect for SDH layer.

ECD, ECD is selected as defect for SDH layer.

EPD, EPD is selected as defect for SDH layer.

B3, B3 is selected as defect for SDH layer.

HPRei, HPRei is selected as defect for SDH layer.

TUAis, TUAis is selected as defect for SDH layer.

TULop, TULop is selected as defect for SDH layer.

LPRFi, LPRFi is selected as defect for SDH layer.

LPRDi, LPRDi is selected as defect for SDH layer.

ESDLop, ESDLop is selected as defect for SDH layer.

ECDLop, ECDLop is selected as defect for SDH layer.

EPDLop, EPDLop is selected as defect for SDH layer.

BIP2, BIP2 is selected as defect for SDH layer.

LPREi, LPREi is selected as defect for SDH layer.

E1AIS, E1AIS is selected as defect for SDH layer.

E1RAI, E1RAI is selected as defect for SDH layer.

E1LOF, E1LOF is selected as defect for SDH layer.

RAIMf, RAIMf is selected as defect for SDH layer.

LOMF, LOMF is selected as defect for SDH layer.

TS16AIS, TS16AIS is selected as defect for SDH layer.

E1FAS, E1FAS is selected as defect for SDH layer.

E2AIS, E2AIS is selected as defect for SDH layer.

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

E2RAI, E2RAI is selected as defect for SDH layer.
E2LOF, E2LOF is selected as defect for SDH layer.
E2FAS, E2FAS is selected as defect for SDH layer.
E3AIS, E3AIS is selected as defect for SDH layer.
E3RAI, E3RAI is selected as defect for SDH layer.
E3LOF, E3LOF is selected as defect for SDH layer.
E3FAS, E3FAS is selected as defect for SDH layer.
E4AIS, E4AIS is selected as defect for SDH layer.
E4RAI, E4RAI is selected as defect for SDH layer.
E4LOF, E4LOF is selected as defect for SDH layer.
E4FAS, DS1AIS is selected as defect for SDH layer.
DS1AIS, DS1AIS is selected as defect for SDH layer.
RAI, RAI is selected as defect for SDH layer.
DS1OOF, DS1OOF is selected as defect for SDH layer.
FRAMingbit, Framing bit is selected as defect for SDH layer.
CRC6, CRC6 is selected as defect for SDH layer.
DS3AIS, DS3AIS is selected as defect for SDH layer.
RDI, RDI is selected as defect for SDH layer.
DS3OOF, DS3OOF is selected as defect for SDH layer.
IDLE, IDLE is selected as defect for SDH layer.
CBIT, CBIT is selected as defect for SDH layer.
FBIT, FBIT is selected as defect for SDH layer.
PBIT, PBIT is selected as defect for SDH layer.
FEBE, FEBE is selected as defect for SDH layer.
PLOSs, Pattern Loss s is selected as defect for SDH layer.
BITerror, Bit error is selected as defect for SDH layer.

**:SENSe[1..n]:DATA:TELEcom:SDT:SDH:
DSElection?****Example(s)**

- * SENS:DATA:TEL:SDT:SDH:LAY:TYPE PORT
- * SENS:DATA:TEL:SDT:SDH:DSEL LOS
- * SENS:DATA:TEL:SDT:SDH:DSEL? Returns LOS

See Also

- * SENSe[1..n]:DATA:TELEcom:SDT:SDH:LAYer:TYPE
 - * SENSe[1..n]:DATA:TELEcom:SDT:SDH:DSElection
 - * SENSe[1..n]:DATA:TELEcom:SDT
-

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

Description	<p>This command selects on which layer the service disruption time test will be performed for PDH.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SDT:PDH:LAYer: TYPE<wsp>PORTE E1 E2 E3 E4 PATtern</p>
Parameter(s)	<p>Layer: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PORTE E1 E2 E3 E4 PATtern. Selects on which layer the service disruption time test will be performed.Choices depend on the selected test path. Selects on which layer the service disruption time test will be performed.Choices depend on the selected test path. PORTE, selects the layer on which the service disruption time test will be performed as Port. E1, selects the layer on which the service disruption time test will be performed as E1. E2, selects the layer on which the service disruption time test will be performed as E2.</p>

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

E3, selects the layer on which the service disruption time test will be performed as E3.
E4, selects the layer on which the service disruption time test will be performed as E4.
PATTern, selects the layer on which the service disruption time test will be performed as Pattern.

Example(s)

* SENS:DATA:TEL:SDT:PDH:LAY:TYPE PORT
* SENS:DATA:TEL:SDT:PDH:LAY:TYPE?
Returns PORTE

See Also

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

**:SENSE[1..n]:DATA:TELEcom:SDT:PDH:LAYER:
TYPE?**

Description	<p>This query returns the layer for service disruption time test for PDH.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSE[1..n]:DATA:TELEcom:SDT:PDH:LAYER: TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Layer></p>
Response(s)	<p>Layer: The response data syntax for <Layer> is defined as a <CHARACTER RESPONSE DATA> element. Returns the layer for service disruption time test will be performed. Returns the layer for service disruption time test will be performed. PORTE, selects the layer on which the service disruption time test will be performed as Port. E1, selects the layer on which the service disruption time test will be performed as E1. E2, selects the layer on which the service disruption time test will be performed as E2.</p>

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

E3, selects the layer on which the service disruption time test will be performed as E3.
E4, selects the layer on which the service disruption time test will be performed as E4.
PATTERN, selects the layer on which the service disruption time test will be performed as Pattern.

Example(s)

* SENS:DATA:TEL:SDT:PDH:LAY:TYPE PORT
* SENS:DATA:TEL:SDT:PDH:LAY:TYPE?
Returns PORTE

See Also

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

:SENSe[1..n]:DATA:TELEcom:SDT:PDH: DSElection

Description	<p>This command selects the defect of PDH for specific layer.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDT:PDH: DSElection<wsp>LOSE E1AIS E1RAI E1LOF R AIMf LOMF TS16AIS E1FAS E2AIS E2RAI E2LO F E2FAS E3AIS E3RAI E3LOF E3FAS E4AIS E4RAI E4LOF E4FAS PLOSs BITerror</pre>
Parameter(s)	<p>Selection:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>LOSE E1AIS E1RAI E1LOF RAIMf LOMF TS16A IS E1FAS E2AIS E2RAI E2LOF E2FAS E3AIS E3RAI E3LOF E3FAS E4AIS E4RAI E4LOF E4FAS PLOSs BITerror.</pre> <p>Selects the defect for PDH layer.</p> <p>LOSE, selects the defect for PDH layer as LOS.</p> <p>E1AIS, selects the defect for PDH layer as AIS.</p> <p>E1RAI, selects the defect for PDH layer as RAI.</p> <p>E1LOF, selects the defect for PDH layer as LOF.</p> <p>RAIMf, selects the defect for PDH layer as RAIMF.</p> <p>LOMF, selects the defect for PDH layer as LOMF.</p>

:SENSe[1..n]:DATA:TELEcom:SDT:PDH: DSElection

TS16AIS, selects the defect for PDH layer as TS16AIS.

E1FAS, selects the defect for PDH layer as FAS.

E2AIS, selects the defect for PDH layer as AIS.

E2RAI, selects the defect for PDH layer as RAI.

E2LOF, selects the defect for PDH layer as LOF.

E2FAS, selects the defect for PDH layer as FAS.

E3AIS, selects the defect for PDH layer as AIS.

E3RAI, selects the defect for PDH layer as RAI.

E3LOF, selects the defect for PDH layer as LOF.

E3FAS, selects the defect for PDH layer as FAS.

E4AIS, selects the defect for PDH layer as AIS.

E4RAI, selects the defect for PDH layer as RAI.

E4LOF, selects the defect for PDH layer as LOF.

E4FAS, selects the defect for PDH layer as FAS.

PLOSSs, selects the defect for PDH layer as Pattern Loss.

BITerror, selects the defect for PDH layer as Bit Error.

Example(s)

* SENS:DATA:TEL:SDT:PDH:LAY:TYPE PORT

* SENS:DATA:TEL:SDT:PDH:DSEL LOSE

* SENS:DATA:TEL:SDT:PDH:DSEL? Returns LOSE

See Also

* SENSE[1..n]:DATA:TELEcom:SDT:PDH:LAYer:
TYPE

* SENSE[1..n]:DATA:TELEcom:SDT:PDH:
DSElection?

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

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

Description	This query returns the defect of PDH for specific layer. At *RST, this value is device dependent.
Syntax	:SENSe[1..n]:DATA:TELEcom:SDT:PDH: DSElection?
Parameter(s)	None
Response Syntax	<Selection>

**:SENSe[1..n]:DATA:TELecom:SDT:PDH:
DSElection?****Response(s)**

Selection:

The response data syntax for <Selection> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the defect for PDH layer.

LOSE, selects the defect for PDH layer as LOS.

E1AIS, selects the defect for PDH layer as AIS.

E1RAI, selects the defect for PDH layer as RAI.

E1LOF, selects the defect for PDH layer as LOF.

RAIMF, selects the defect for PDH layer as RAIMF.

LOMF, selects the defect for PDH layer as LOMF.

TS16AIS, selects the defect for PDH layer as TS16AIS.

E1FAS, selects the defect for PDH layer as FAS.

E2AIS, selects the defect for PDH layer as AIS.

E2RAI, selects the defect for PDH layer as RAI.

:SENSe[1..n]:DATA:TELecom:SDT:PDH: DSElection?

E2LOF, selects the defect for PDH layer as LOF.
E2FAS, selects the defect for PDH layer as FAS.
E3AIS, selects the defect for PDH layer as AIS.
E3RAI, selects the defect for PDH layer as RAI.
E3LOF, selects the defect for PDH layer as LOF.
E3FAS, selects the defect for PDH layer as FAS.
E4AIS, selects the defect for PDH layer as AIS.
E4RAI, selects the defect for PDH layer as RAI.
E4LOF, selects the defect for PDH layer as LOF.
E4FAS, selects the defect for PDH layer as FAS.
PLOSS, selects the defect for PDH layer as Pattern Loss.
BITERROR, selects the defect for PDH layer as Bit Error.

Example(s)

* SENS:DATA:TEL:SDT:PDH:LAY:TYPE PORT
* SENS:DATA:TEL:SDT:PDH:DSEL LOSE
* SENS:DATA:TEL:SDT:PDH:DSEL? Returns LOSE

See Also

* SENSe[1..n]:DATA:TELecom:SDT:PDH:LAYer:
TYPE
* SENSe[1..n]:DATA:TELecom:SDT:PDH:
DSElection
* SENSe[1..n]:DATA:TELecom:SDT

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

Description	<p>This command selects on which layer the service disruption time test will be performed for DSN.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDT:DSN:LAYer: TYPE<wsp>PORTe DS1 DS3 PATtern</pre>
Parameter(s)	<p>Layer:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PORTe DS1 DS3 PATtern.</p> <p>Selects on which layer the service disruption time test will be performed. Choices depend on the selected test path.</p> <p>PORTe, selects the layer on which the service disruption time test will be performed as Port.</p> <p>DS1, selects the layer on which the service disruption time test will be performed as DS1.</p> <p>DS3, selects the layer on which the service disruption time test will be performed as DS3.</p>

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

PATtern, selects the layer on which the service disruption time test will be performed as Pattern.

Example(s)

* SENS:DATA:TEL:SDT:DSN:LAY:TYPE PORT
* SENS:DATA:TEL:SDT:DSN:LAY:TYPE?
Returns PORT

See Also

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

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

Description	This query returns the layer for service disruption time test for DSN. At *RST, this value is device dependent.
Syntax	:SENSe[1..n]:DATA:TELEcom:SDT:DSN:LAYer: TYPE?
Parameter(s)	None
Response Syntax	<Layer>

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

Response(s)	<p>Layer:</p> <p>The response data syntax for <Layer> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the layer for service disruption time test will be performed.</p> <p>PORTE, selects the layer on which the service disruption time test will be performed as Port.</p> <p>DS1, selects the layer on which the service disruption time test will be performed as DS1.</p> <p>DS3, selects the layer on which the service disruption time test will be performed as DS3.</p> <p>PATTERN, selects the layer on which the service disruption time test will be performed as Pattern.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:DSN:LAY:TYPE PORT* SENS:DATA:TEL:SDT:DSN:LAY:TYPE? <p>Returns PORT</p>
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:SDT:DSN:LAYer:TYPE* SENSe[1..n]:DATA:TELEcom:SDT:DSN:DSElection* SENSe[1..n]:DATA:TELEcom:SDT

:SENSe[1..n]:DATA:TELEcom:SDT:DSN: DSElection

Description	<p>This command selects the defect of DSN for specific layer.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDT:DSN: DSElection<wsp>LOSE BPV EXZ DS1AIS RAI DS1OOF FRAMingbit CRC6 DS3AIS RDI DS3OOF IDLE CBIT FBIT PBIT FEBE PLOSs BITerror</pre>
Parameter(s)	<p>Selection:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>LOSE BPV EXZ DS1AIS RAI DS1OOF FRAMingbit CRC6 DS3AIS RDI DS3OOF IDLE CBIT FBIT PBIT FEBE PLOSs BITerror.</pre> <p>Selects the defect for DSN layer.</p> <p>LOSE, selects the defect for DSN layer as LOS.</p> <p>BPV, selects the defect for DSN layer as BPV.</p> <p>EXZ, selects the defect for DSN layer as EXZ.</p> <p>DS1AIS, selects the defect for DSN layer as AISL.</p> <p>RAI, selects the defect for DSN layer as RAI.</p> <p>DS1OOF, selects the defect for DSN layer as SEF.</p>

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

FRAMingbit, selects the defect for DSN layer as Framing Bit.

CRC6, selects the defect for DSN layer as CRC6.

DS3AIS, selects the defect for DSN layer as AIS.

RDI, selects the defect for DSN layer as RDI.

DS3OOF, selects the defect for DSN layer as SEF.

IDLE, selects the defect for DSN layer as IDLE.

CBIT, selects the defect for DSN layer as CBIT.

FBIT, selects the defect for DSN layer as FBIT.

PBIT, selects the defect for DSN layer as PBIT.

FEBE, selects the defect for DSN layer as FEBE.

PLOSS, selects the defect for DSN layer as Pattern Loss.

BITerror, selects the defect for DSN layer as Bit Error.

Example(s)

- * SENS:DATA:TEL:SDT:DSN:LAY:TYPE PORT
- * SENS:DATA:TEL:SDT:DSN:DSEL LOSE
- * SENS:DATA:TEL:SDT:DSN:DSEL? Returns LOSE

See Also

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

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

Description	<p>This query returns the defect of DSN for specific layer.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:SDT:DSN: DSElection?
Parameter(s)	None
Response Syntax	<Selection>
Response(s)	<p>Selection:</p> <p>The response data syntax for <Selection> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the defect for DSN layer.</p> <p>LOSE, selects the defect for DSN layer as LOS.</p> <p>BPV, selects the defect for DSN layer as BPV.</p> <p>EXZ, selects the defect for DSN layer as EXZ.</p> <p>DS1AIS, selects the defect for DSN layer as AISL.</p> <p>RAI, selects the defect for DSN layer as RAI.</p> <p>DS1OOF, selects the defect for DSN layer as SEF.</p> <p>FRAMINGBIT, selects the defect for DSN layer as Framing Bit.</p> <p>CRC6, selects the defect for DSN layer as CRC6.</p>

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

DS3AIS, selects the defect for DSN layer as AIS.
RDI, selects the defect for DSN layer as RDI.
DS3OOF, selects the defect for DSN layer as SEF.
IDLE, selects the defect for DSN layer as IDLE.
CBIT, selects the defect for DSN layer as CBIT.
FBIT, selects the defect for DSN layer as FBIT.
PBIT, selects the defect for DSN layer as PBIT.
FEBE, selects the defect for DSN layer as FEBE.
PLOSS, selects the defect for DSN layer as Pattern Loss.
BITERROR, selects the defect for DSN layer as Bit Error.

Example(s)

* SENS:DATA:TEL:SDT:DSN:LAY:TYPE PORT
* SENS:DATA:TEL:SDT:DSN:DSEL LOSE
* SENS:DATA:TEL:SDT:DSN:DSEL? Returns LOSE

See Also

* SENSe[1..n]:DATA:TELEcom:SDT:DSN:LAY:
TYPE
* SENSe[1..n]:DATA:TELEcom:SDT:DSN:
DSElection
* SENSe[1..n]:DATA:TELEcom:SDT

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

This command selects the period without any defects before stopping SDT measurement.

At *RST, this value is device dependent.

Syntax

:SENSe[1..n]:DATA:TELEcom:SDT:NDTime
<wsp><Time> | MAXimum | MINimum |
MAXimum | MINimum

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

Parameter(s)	<p>Time:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Selects the period without any defects before stopping SDT measurement. Choices are from 10 μs to 99990 μs.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:NDT 15000* SENS:DATA:TEL:SDT:NDT? Returns 15000
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE* SENSe[1..n]:DATA:TELEcom:SDT:SONet:DSElection* SENSe[1..n]:DATA:TELEcom:SDT:NDTime?* SENSe[1..n]:DATA:TELEcom:SDT

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

Description	<p>This query returns the period without any defects before stopping SDT measurement.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDT:NDTime? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period without any defects will be returned.</p>
Response Syntax	<pre><Time></pre>

:SENSe[1..n]:DATA:TELecom:SDT:NDTime?

Response(s)	<p>Time:</p> <p>The response data syntax for <Time> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the period without any defects before stopping SDT measurement.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:NDT 15000* SENS:DATA:TEL:SDT:NDT? Returns 15000
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELecom:SDT:SONet:LAYer:TYPE* SENSe[1..n]:DATA:TELecom:SDT:SONet:DSElection* SENSe[1..n]:DATA:TELecom:SDT:SONet:NDTime* SENSe[1..n]:DATA:TELecom:SDT

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

This command selects the period of time used to calculate the SDT measurements.

At *RST, this value is device dependent.

Syntax

:SENSe[1..n]:DATA:TELEcom:SDT:TPERiod
<wsp><Period> | MAXimum | MINimum

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

Parameter(s)	<p>Period:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Selects the period of time used to calculate the SDT measurement. Choices are from 20 μs to 5 min.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:TPER 15000* SENS:DATA:TEL:SDT:TPER? Returns 15000
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE* SENSe[1..n]:DATA:TELEcom:SDT:SONet:DSElection* SENSe[1..n]:DATA:TELEcom:SDT:NDTime* SENSe[1..n]:DATA:TELEcom:SDT:TPERiod* SENSe[1..n]:DATA:TELEcom:SDT

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

Description	<p>This query returns the period of time used to calculate the SDT measurements.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDT:TPERiod? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value. MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current period of time used to calculate the SDT measurement will be returned.</p>
Response Syntax	<pre><Period></pre>

:SENSe[1..n]:DATA:TELecom:SDT:TPERiod?

Response(s)	<p>Period:</p> <p>The response data syntax for <Period> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the period of time used to calculate the SDT measurements.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:TPER 15000* SENS:DATA:TEL:SDT:TPER? Returns 15000
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELecom:SDT:SONet:LAYer:TYPE* SENSe[1..n]:DATA:TELecom:SDT:SONet:DSElection* SENSe[1..n]:DATA:TELecom:SDT:NDTime* SENSe[1..n]:DATA:TELecom:SDT:TPERiod?* SENSe[1..n]:DATA:TELecom:SDT

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

Description	<p>This command enables or disables the disruption time measurements.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:SDT<wsp><Set>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the disruption time measurements.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT* SENS:DATA:TEL:SDT:SON:DSEL LOS* SENS:DATA:TEL:SDT ON* SENS:DATA:TEL:SDT? Returns 1
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE* SENSe[1..n]:DATA:TELEcom:SDT:SONet:DSELection* SENSe[1..n]:DATA:TELEcom:SDT:NDTime* SENSe[1..n]:DATA:TELEcom:SDT:TPERiod* SENSe[1..n]:DATA:TELEcom:SDT

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

Description	This query returns the status of disruption time measurements. At *RST, this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:SDT?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the disruption time measurements.
Example(s)	* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT * SENS:DATA:TEL:SDT:SON:DSEL LOS * SENS:DATA:TEL:SDT ON * SENS:DATA:TEL:SDT? Returns 1
See Also	* SENSe[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE * SENSe[1..n]:DATA:TELEcom:SDT:SONet:DSELection * SENSe[1..n]:DATA:TELEcom:SDT:NDTime * SENSe[1..n]:DATA:TELEcom:SDT:TPERiod * SENSe[1..n]:DATA:TELEcom:SDT

:FETCh[1..n]:DATA:TELEcom:SDT:SHORtest?

Description	This query returns the shortest disruption time. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:SHORtest?
Parameter(s)	None
Response Syntax	<Shortest>

:FETCh[1..n]:DATA:TELEcom:SDT:SHORTest?

Response(s)	<p>Shortest:</p> <p>The response data syntax for <Shortest> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the shortest disruption time.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT* SENS:DATA:TEL:SDT:SON:DSEL LOS* SENS:DATA:TEL:SDT ON* SENS:DATA:TEL:SDT:SHORT? <p>Returns the shortest disruption time.</p>
See Also	<ul style="list-style-type: none">* SENSE[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE* SENSE[1..n]:DATA:TELEcom:SDT:SONet:DSELection* SENSE[1..n]:DATA:TELEcom:SDT:NDTime* SENSE[1..n]:DATA:TELEcom:SDT:TPERiod* SENSE[1..n]:DATA:TELEcom:SDT?* FETCh[1..n]:DATA:TELEcom:SDT:SHORTest?

:FETCh[1..n]:DATA:TELEcom:SDT:LONGest?

Description	This query returns the longest disruption time. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:LONGest?
Parameter(s)	None
Response Syntax	<Longest>

:FETCh[1..n]:DATA:TELEcom:SDT:LONGest?

Response(s)	Longest: The response data syntax for <Longest> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the longest disruption time.
Example(s)	* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT * SENS:DATA:TEL:SDT:SON:DSEL LOS * SENS:DATA:TEL:SDT ON * SENS:DATA:TEL:SDT:LONG? Returns the longest disruption time.
See Also	* SENSE[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE * SENSE[1..n]:DATA:TELEcom:SDT:SONet:DSELection * SENSE[1..n]:DATA:TELEcom:SDT:NDTime * SENSE[1..n]:DATA:TELEcom:SDT:TPERiod * SENSE[1..n]:DATA:TELEcom:SDT? * FETCh[1..n]:DATA:TELEcom:SDT:LONGest?

:FETCh[1..n]:DATA:TELEcom:SDT:LAST?

Description	This query returns the length of the last disruption time. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:LAST?
Parameter(s)	None
Response Syntax	<Last>

:FETCh[1..n]:DATA:TELEcom:SDT:LAST?

Response(s)	<p>Last:</p> <p>The response data syntax for <Last> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the length of the last disruption time.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT* SENS:DATA:TEL:SDT:SON:DSEL LOS* SENS:DATA:TEL:SDT ON* SENS:DATA:TEL:SDT:LAST? <p>Returns the length of the last disruption time.</p>
See Also	<ul style="list-style-type: none">* SENSE[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE* SENSE[1..n]:DATA:TELEcom:SDT:SONet:DSELection* SENSE[1..n]:DATA:TELEcom:SDT:NDTime* SENSE[1..n]:DATA:TELEcom:SDT:TPERiod* SENSE[1..n]:DATA:TELEcom:SDT?* FETCh[1..n]:DATA:TELEcom:SDT:LAST?

:FETCh[1..n]:DATA:TELEcom:SDT:AVERAge?

Description	This query returns the average disruption time. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:AVERAge?
Parameter(s)	None
Response Syntax	<Average>

:FETCh[1..n]:DATA:TELEcom:SDT:AVERAge?

Response(s)	<p>Average: The response data syntax for <Average> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the average disruption time.</p>
Example(s)	<p>* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT * SENS:DATA:TEL:SDT:SON:DSEL LOS * SENS:DATA:TEL:SDT ON * SENS:DATA:TEL:SDT:AVER? Returns the average disruption time.</p>
See Also	<p>* SENSE[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE * SENSE[1..n]:DATA:TELEcom:SDT:SONet:DSELection * SENSE[1..n]:DATA:TELEcom:SDT:NDTime * SENSE[1..n]:DATA:TELEcom:SDT:TPERiod * SENSE[1..n]:DATA:TELEcom:SDT? * FETCh[1..n]:DATA:TELEcom:SDT:AVERAge?</p>

:FETCh[1..n]:DATA:TELEcom:SDT:TOTal?

Description	This query returns the total disruption time. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:TOTal?
Parameter(s)	None
Response Syntax	<Total>

:FETCh[1..n]:DATA:TELEcom:SDT:TOTal?

Response(s)	Total: The response data syntax for <Total> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the total disruption time.
Example(s)	* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT * SENS:DATA:TEL:SDT:SON:DSEL LOS * SENS:DATA:TEL:SDT ON * SENS:DATA:TEL:SDT:TOT? Returns the total disruption time.
See Also	* SENSE[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE * SENSE[1..n]:DATA:TELEcom:SDT:SONet:DSELection * SENSE[1..n]:DATA:TELEcom:SDT:NDTime * SENSE[1..n]:DATA:TELEcom:SDT:TPERiod * SENSE[1..n]:DATA:TELEcom:SDT? * FETCh[1..n]:DATA:TELEcom:SDT:TOTal?

**:FETCh[1..n]:DATA:TELEcom:SDT:ALARm:
HISTory?**

Description	This query returns the history status of service disruption measurements. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:ALARm: HISTory?
Parameter(s)	None
Response Syntax	<History>

**:FETCh[1..n]:DATA:TELEcom:SDT:ALARM:
HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of service disruption measurements.</p> <p>PRESENT, indicates if any SDT occurred in the past.</p> <p>ABSENT, indicates if no SDT occurred in the past.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT* SENS:DATA:TEL:SDT:SON:DSEL LOS* SENS:DATA:TEL:SDT ON* SENS:DATA:TEL:SDT:ALARM:HIST? <p>Returns the history status of service disruption measurements.</p>
See Also	<ul style="list-style-type: none">* SENSE[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE* SENSE[1..n]:DATA:TELEcom:SDT:SONet:DSELection* SENSE[1..n]:DATA:TELEcom:SDT:NDTime* SENSE[1..n]:DATA:TELEcom:SDT:TPERiod* SENSE[1..n]:DATA:TELEcom:SDT?* FETCh[1..n]:DATA:TELEcom:SDT:ALARM:HISTory?

**:FETCh[1..n]:DATA:TELEcom:SDT:ALARm:
CURRent?**

Description	This query returns the current status of service disruption measurements. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:ALARm: CURRent?
Parameter(s)	None
Response Syntax	<Current>

:FETCh[1..n]:DATA:TELEcom:SDT:ALARM:CURRENT?

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of service disruption measurements.</p> <p>PRESENT, indicates when there is no SDT.</p> <p>ABSENT, indicates if there is a SDT, and last until the next No Defect Time has been met or the test period is elapsed.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT* SENS:DATA:TEL:SDT:SON:DSEL LOS* SENS:DATA:TEL:SDT ON* SENS:DATA:TEL:SDT:ALARM:CURR? <p>Returns the current status of service disruption measurements.</p>
See Also	<ul style="list-style-type: none">* SENSE[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE* SENSE[1..n]:DATA:TELEcom:SDT:SONet:DSELection* SENSE[1..n]:DATA:TELEcom:SDT:NDTime* SENSE[1..n]:DATA:TELEcom:SDT:TPERiod* SENSE[1..n]:DATA:TELEcom:SDT?* FETCh[1..n]:DATA:TELEcom:SDT:ALARM:CURRENT?

**:FETCh[1..n]:DATA:TELEcom:SDT:ALARm:
SECOnds?**

Description	This query returns the number of seconds within which service disruption measurements occurred. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:ALARm: SECOnds?
Parameter(s)	None
Response Syntax	<Seconds>

:FETCh[1..n]:DATA:TELEcom:SDT:ALARm:SECOnds?

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds within which service disruption measurements occurred.
Example(s)	* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT * SENS:DATA:TEL:SDT:SON:DSEL LOS * SENS:DATA:TEL:SDT ON * SENS:DATA:TEL:SDT:ALARm:SEC? Returns the number of seconds within which service disruption measurements occurred.
See Also	* SENSE[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE * SENSE[1..n]:DATA:TELEcom:SDT:SONet:DSELection * SENSE[1..n]:DATA:TELEcom:SDT:NDTime * SENSE[1..n]:DATA:TELEcom:SDT:TPERiod * SENSE[1..n]:DATA:TELEcom:SDT? * FETCh[1..n]:DATA:TELEcom:SDT:ALARm:SECOnds?

:FETCh[1..n]:DATA:TELEcom:SDT:COUNT?

Description	This query returns the number of service disruption counts that happened since the beginning of the Service Disruption Time (SDT) test. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:COUNT?
Parameter(s)	None
Response Syntax	<Count>

:FETCh[1..n]:DATA:TELEcom:SDT:COUNt?

Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of service disruption counts that happened since the beginning of the SDT test.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:SDT:SON:LAY:TYPE PORT* SENS:DATA:TEL:SDT:SON:DSEL LOS* SENS:DATA:TEL:SDT ON* SENS:DATA:TEL:SDT:COUN? <p>Returns the number of service disruption counts.</p>
See Also	<ul style="list-style-type: none">* SENSE[1..n]:DATA:TELEcom:SDT:SONet:LAYer:TYPE* SENSE[1..n]:DATA:TELEcom:SDT:SONet:DSELection* SENSE[1..n]:DATA:TELEcom:SDT:NDTime* SENSE[1..n]:DATA:TELEcom:SDT:TPERiod* SENSE[1..n]:DATA:TELEcom:SDT?* FETCh[1..n]:DATA:TELEcom:SDT:COUNt?

:SENSe[1..n]:DATA:TELEcom:RTD:MODE

Description	<p>This command selects the Round Trip Delay (RTD) test mode.</p> <p>At *RST, this value is set to SINGLE.</p>
Syntax	<code>:SENSe[1..n]:DATA:TELEcom:RTD:MODE<wsp> SINGLE CONTInuous</code>
Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SINGLE CONTInuous.</p> <p>Selects the round trip delay test mode. SINGLE, selects the round trip delay as single. CONTInuous, selects the round trip delay as continuously in a repetitive manner.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:RTD:MODE SING* SENS:DATA:TEL:RTD:MODE? Returns SINGLE
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:RTD:MODE?* SENSe[1..n]:DATA:TELEcom:RTD

:SENSe[1..n]:DATA:TELEcom:RTD:MODE?

Description	This query returns the Round Trip Delay (RTD) test mode. At *RST, this value is set to SINGLE.
Syntax	:SENSe[1..n]:DATA:TELEcom:RTD:MODE?
Parameter(s)	None
Response Syntax	<Mode>
Response(s)	Mode: The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element. Returns the round trip delay test mode. SINGLE, single mode is selected as round trip delay. CONTINUOUS, continuous mode is selected as round trip delay.
Example(s)	* SENS:DATA:TEL:RTD:MODE SING * SENS:DATA:TEL:RTD:MODE? Returns SINGLE
See Also	* SENSE[1..n]:DATA:TELEcom:RTD:MODE * SENSE[1..n]:DATA:TELEcom:RTD

:SENSe[1..n]:DATA:TELEcom:RTD

Description	<p>This command enables or disables the Round Trip Delay (RTD) measurements.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:RTD<wsp><Set>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the round trip delay measurements.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:RTD:MODE SING* SENS:DATA:TEL:RTD ON* SENS:DATA:TEL:RTD? Returns 1
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELEcom:RTD:MODE* SENSe[1..n]:DATA:TELEcom:RTD?

:SENSe[1..n]:DATA:TELEcom:RTD?

Description This query returns the status of Round Trip Delay (RTD) measurements.

At *RST, this value is set to OFF.

Syntax :SENSe[1..n]:DATA:TELEcom:RTD?

Parameter(s) None

Response Syntax <Set>

Response(s) Set:
The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the status of round trip delay measurements.

Example(s) * SENS:DATA:TEL:RTD:MODE SING
* SENS:DATA:TEL:RTD ON
* SENS:DATA:TEL:RTD? Returns 1

See Also * SENSe[1..n]:DATA:TELEcom:RTD:MODE
* SENSe[1..n]:DATA:TELEcom:RTD

**:FETCh[1..n]:DATA:TELEcom:RTD:DELay:
STATUs?**

Description	This query returns the test status of the Round Trip Delay (RTD) measurements. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:RTD:DELay: STATUs?
Parameter(s)	None
Response Syntax	<Status>

**:FETCh[1..n]:DATA:TELEcom:RTD:DELAy:
STATUs?**

Response(s)

Status:

The response data syntax for <Status> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the test status of the round trip delay measurements.

READY, Ready indicates that the test is ready to perform RTD measurement.

RUNNING, Running indicates that the RTD test is running.

CANCELLED, Cancelled indicates that the RTD test has been stopped before its completion.

CFAILED, Calibration Failed indicates that the test calibration failed.

Example(s)

* SENS:DATA:TEL:RTD:MODE SING

* SENS:DATA:TEL:RTD ON

* FETC:DATA:TEL:RTD:DEL:STAT?

Returns the test status of RTD.

See Also

* SENSE[1..n]:DATA:TELEcom:RTD:MODE

* SENSE[1..n]:DATA:TELEcom:RTD

:FETCh[1..n]:DATA:TELEcom:RTD:DELay:LAST?

Description	This query returns the result of the last Round Trip Delay measurement. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:RTD:DELay:LAST?
Parameter(s)	None
Response Syntax	<Last>
Response(s)	Last: The response data syntax for <Last> is defined as a <STRING RESPONSE DATA> element. Returns the result of the last round trip delay measurement.
Example(s)	* SENS:DATA:TEL:RTD:MODE SING * SENS:DATA:TEL:RTD ON * FETC:DATA:TEL:RTD:DEL:LAST? Returns the result of the last round trip delay measurement.
See Also	* SENSE[1..n]:DATA:TELEcom:RTD:MODE * SENSE[1..n]:DATA:TELEcom:RTD

**:FETCh[1..n]:DATA:TELEcom:RTD:DELay:
MAXimum?**

Description	<p>This query returns the maximum Round Trip Delay recorded.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:RTD:DELay: MAXimum?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Maximum></p>
Response(s)	<p>Maximum: The response data syntax for <Maximum> is defined as a <STRING RESPONSE DATA> element. Returns the maximum round trip delay recorded.</p>
Example(s)	<p>* SENS:DATA:TEL:RTD:MODE SING * SENS:DATA:TEL:RTD ON * FETC:DATA:TEL:RTD:DEL:MAX? Returns the maximum Round Trip Delay recorded.</p>
See Also	<p>* SENSE[1..n]:DATA:TELEcom:RTD:MODE * SENSE[1..n]:DATA:TELEcom:RTD</p>

**:FETCh[1..n]:DATA:TELEcom:RTD:DELAy:
MINimum?**

Description	<p>This query returns the minimum Round Trip Delay recorded.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:RTD:DELAy: MINimum?
Parameter(s)	None
Response Syntax	<Minimum>
Response(s)	<p>Minimum: The response data syntax for <Minimum> is defined as a <STRING RESPONSE DATA> element. Returns the minimum round trip delay recorded.</p>
Example(s)	<p>* SENS:DATA:TEL:RTD:MODE SING * SENS:DATA:TEL:RTD ON * FETC:DATA:TEL:RTD:DEL:MIN? Returns the minimum Round Trip Delay recorded.</p>
See Also	<p>* SENSE[1..n]:DATA:TELEcom:RTD:MODE * SENSE[1..n]:DATA:TELEcom:RTD</p>

:FETCh[1..n]:DATA:TELEcom:RTD:DELay: AVERAge?

Description	<p>This query returns the average Round Trip Delay measurements.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:RTD:DELay: AVERAge?</code>
Parameter(s)	None
Response Syntax	<code><Average></code>
Response(s)	<p>Average: The response data syntax for <code><Average></code> is defined as a <code><STRING RESPONSE DATA></code> element. Returns the average round trip delay measurements.</p>
Example(s)	<pre>* SENS:DATA:TEL:RTD:MODE SING * SENS:DATA:TEL:RTD ON * FETC:DATA:TEL:RTD:DEL:AVER? Returns the average Round Trip Delay measurements.</pre>
See Also	<pre>* SENSE[1..n]:DATA:TELEcom:RTD:MODE * SENSE[1..n]:DATA:TELEcom:RTD</pre>

**:FETCh[1..n]:DATA:TELEcom:RTD:COUNT:
SUCCEssful?**

Description	<p>This query returns the total number of successful Round Trip Delay measurements.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:RTD:COUNT: SUCCEssful?
Parameter(s)	None
Response Syntax	<Successful>
Response(s)	<p>Successful:</p> <p>The response data syntax for <Successful> is defined as a <NR2 NUMERICAL RESPONSE DATA> element.</p> <p>Returns the total number of successful round trip delay measurements.</p>
Example(s)	<p>* SENS:DATA:TEL:RTD:MODE SING * SENS:DATA:TEL:RTD ON * FETC:DATA:TEL:RTD:COUN:SUC? Returns the total number of successful measurements.</p>
See Also	<p>* SENSE[1..n]:DATA:TELEcom:RTD:MODE * SENSE[1..n]:DATA:TELEcom:RTD</p>

:FETCh[1..n]:DATA:TELEcom:RTD:COUNT: FAILed?

Description	<p>This query returns the total number of failed Round Trip Delay measurements.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:RTD:COUNT: FAILed?</code>
Parameter(s)	None
Response Syntax	<Failed>
Response(s)	<p>Failed:</p> <p>The response data syntax for <Failed> is defined as a <NR2 NUMERICAL RESPONSE DATA> element.</p> <p>Returns the total number of failed round trip delay measurements.</p>
Example(s)	<pre>* SENS:DATA:TEL:RTD:MODE SING * SENS:DATA:TEL:RTD ON * FETC:DATA:TEL:RTD:COUN:FAIL? Returns the total number of failed measurements.</pre>
See Also	<pre>* SENSE[1..n]:DATA:TELEcom:RTD:MODE * SENSE[1..n]:DATA:TELEcom:RTD</pre>

:SENSe[1..n]:DATA:TELEcom:RTD:RESet

Description	<p>This command resets the results and measurement counts of Round Trip Delay.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:RTD:RESet
Parameter(s)	None
Example(s)	* SENS:DATA:TEL:RTD:MODE SING * SENS:DATA:TEL:RTD ON * SENS:DATA:TEL:RTD:RES
See Also	* SENSe[1..n]:DATA:TELEcom:RTD:MODE * SENSe[1..n]:DATA:TELEcom:RTD?

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

Description

This command selects on which layer the service disruption time test will be performed for OTN.

At *RST, this value is device dependent.

Syntax

```
:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:  
TYPE<wsp>  
PORT|SECTIon|RS|MS|LINE|HOP|LOP|FEC|O  
TU1|OTU2|OTU3|ODU1|ODU2|ODU3|ODU1E|  
ODU2E|OTU1E|OTU2E|PATTerN|OPU0|ODU0
```

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

Parameter(s)

Layer:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

PORT|SECTion|RS|MS|LINE|HOP|LOP|FEC|OTU1|OTU2|OTU3|ODU1|ODU2|ODU3|ODU1E|ODU2E|OTU1E|OTU2E|PATtern|OPU0|ODU0.

Selects on which layer the service disruption time test will be performed. Choices depend on the selected test path.

PORT, selects the layer on which the service disruption time test will be performed as Port.

SECTion, selects the layer on which the service disruption time test will be performed as Section.

RS, selects the layer on which the service disruption time test will be performed as RS.

MS, selects the layer on which the service disruption time test will be performed as MS.

LINE, selects the layer on which the service disruption time test will be performed as Line.

HOP, selects the layer on which the service disruption time test will be performed as HOP.

LOP, selects the layer on which the service disruption time test will be performed as LOP.

FEC, selects the layer on which the service disruption time test will be performed as FEC.

OTU1, selects the layer on which the service disruption time test will be performed as OTU1.

OTU2, selects the layer on which the service disruption time test will be performed as OTU2.

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

OTU3, selects the layer on which the service disruption time test will be performed as OTU3.
ODU1, selects the layer on which the service disruption time test will be performed as ODU1.
ODU2, selects the layer on which the service disruption time test will be performed as ODU2.
ODU3, selects the layer on which the service disruption time test will be performed as ODU3.
ODU1E, selects the layer on which the service disruption time test will be performed as ODU1E.
ODU2E, selects the layer on which the service disruption time test will be performed as ODU2E.
OTU1E, selects the layer on which the service disruption time test will be performed as OTU1E.
OTU2E, selects the layer on which the service disruption time test will be performed as OTU2E.
PATTern, selects the layer on which the service disruption time test will be performed as Pattern.
OPU0, selects the layer on which the service disruption time test will be performed as OPU0.
ODU0, selects the layer on which the service disruption time test will be performed as ODU0.

Example(s)

SENS:DATA:TEL:SDT:OTN:LAY:TYPE SECT
SENS:DATA:TEL:SDT:OTN:LAY:TYPE Returns
SECTION

See Also

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

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

Description	This query returns the layer for service disruption time test for OTN. At *RST, this value is device dependent.
Syntax	:SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer: TYPE?
Parameter(s)	None
Response Syntax	<Layer>

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

Response(s)

Layer:

The response data syntax for <Layer> is defined as a <CHARACTER RESPONSE DATA> element. Returns the layer for service disruption time test will be performed.

PORT, selects the layer on which the service disruption time test will be performed as Port.

SECTION, selects the layer on which the service disruption time test will be performed as Section.

RS, selects the layer on which the service disruption time test will be performed as RS.

MS, selects the layer on which the service disruption time test will be performed as MS.

LINE, selects the layer on which the service disruption time test will be performed as Line.

HOP, selects the layer on which the service disruption time test will be performed as HOP.

LOP, selects the layer on which the service disruption time test will be performed as LOP.

FEC, selects the layer on which the service disruption time test will be performed as FEC.

OTU1, selects the layer on which the service disruption time test will be performed as OTU1.

OTU2, selects the layer on which the service disruption time test will be performed as OTU2.

OTU3, selects the layer on which the service disruption time test will be performed as OTU3.

ODU1, selects the layer on which the service disruption time test will be performed as ODU1.

ODU2, selects the layer on which the service disruption time test will be performed as ODU2.

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

ODU3, selects the layer on which the service disruption time test will be performed as ODU3.
 ODU1E, selects the layer on which the service disruption time test will be performed as ODU1E.
 ODU2E, selects the layer on which the service disruption time test will be performed as ODU2E.
 OTU1E, selects the layer on which the service disruption time test will be performed as OTU1E.
 OTU2E, selects the layer on which the service disruption time test will be performed as OTU2E.
 PATtern, selects the layer on which the service disruption time test will be performed as Pattern.
 OPU0, selects the layer on which the service disruption time test will be performed as OPU0.
 ODU0, selects the layer on which the service disruption time test will be performed as ODU0.

Example(s)

```
SENS:DATA:TEL:SDT:OTN:LAY:TYPE SECT
SENS:DATA:TEL:SDT:OTN:LAY:TYPE Returns
SECTION
```

See Also

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

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

Description

This query returns the defect for specific layer for OTN.

At *RST, this value is device dependent

Syntax

:SENSe[1..n]:DATA:TELEcom:SdT:OTN:
DSElection<wsp>
LOS|LOF|SEF|B1|RFAS|AISL|RDIL|B2|REIL|AI
SP|RDIP|EPSD|EPCD|EPPD|LOM|LOPP|PDIP|
B3|REIP|AISV|RDIV|EVSD|EVCD|EVPD|RFIV|L
OPV|BIP2|REIV|OOF|MSAIS|MSRDI|MSREI|AU
AIS|AULOP|H4LOM|HPRDI|ESD|ECD|EPD|HP
REI|TUAIS|TULOP|LPRFI|LPRDI|ESDLop|ECDL
op|EPDLop|LPREI|FCCW|FUCW|FCSYmb|FC
Bit|OAIS|OBDI|OLOF|OOOF|OLOM|OOM|OBI
ae|OIAE|OBIP8|OBEI|FAS|MFAS|ODAIS|ODBD
I|OLCK|OOCI|OFSF|OBSF|OFSD|OBSD|LOFL
om|ODBIP8|ODBEI|PLOSs|BITerror|OPUCSF

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

Parameter(s)

Selection:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

LOS|LOF|SEF|B1|RFAS|AISL|RDIL|B2|REIL|
AISP|RDIP|EPSD|EPCD|EPPD|LOM|LOPP|
PDIP|B3|REIP|AISV|RDIV|EVSD|EVCD|EVPD|
RFIV|LOPV|BIP2|REIV|OOF|MSAIS|MSRDI|
MSREI|AUAIS|AULOP|H4LOM|HPRDI|ESD|ECD
|EPD|HPREI|TUAIS|TULOP|LPRFI|LPRDI|
ESDLop|ECDLop|EPDLop|LPREI|FCCW|FUCW
|FCSYmb|FCBit|OAI|OBDI|OLOF|OOOF|
OLOM|OOM|OBlae|OIAE|OBIP8|OBEI|FAS|
MFAS|ODAIS|ODBDI|OLCK|OOCI|OFSF|OBSF|
OFSd|OBSd|LOFLom|ODBIP8|ODBEI|PLOSS|
BITerror|OPUCSF.

Selects the defect for OTN layer.

LOS, selects the defect for OTN layer as LOS.

LOF, selects the defect for OTN layer as LOF.

SEF, selects the defect for OTN layer as SEF.

B1, selects the defect for OTN layer as B1.

RFAS, selects the defect for OTN layer as RFAS.

AISL, selects the defect for OTN layer as AISL.

RDIL, selects the defect for OTN layer as RDIL.

B2, selects the defect for OTN layer as B2.

REIL, selects the defect for OTN layer as REIL.

AISP, selects the defect for OTN layer as AISP.

RDIP, selects the defect for OTN layer as RDIP.

:SENSE[1..n]:DATA:TELEcom:SDT:OTN: DSElection

EPSD, selects the defect for OTN layer as EPSD.
EPCD, selects the defect for OTN layer as EPCD.
EPPD, selects the defect for OTN layer as EPPD.
LOM, selects the defect for OTN layer as LOM.
LOPP, selects the defect for OTN layer as LOPP.
PDIP, selects the defect for OTN layer as PDIP.
B3, selects the defect for OTN layer as B3.
REIP, selects the defect for OTN layer as REIP.
AISV, selects the defect for OTN layer as AISV.
RDIV, selects the defect for OTN layer as RDIV.
EVSD, selects the defect for OTN layer as EVSD.
EVCD, selects the defect for OTN layer as EVCD.
EVPD, selects the defect for OTN layer as EVPD.
RFIV, selects the defect for OTN layer as RFIV.
LOPV, selects the defect for OTN layer as LOPV.
BIP2, selects the defect for OTN layer as BIP2.
REIV, selects the defect for OTN layer as REIV.
OOF, selects the defect for OTN layer as OOF.
MSAIS, selects the defect for OTN layer as MSAIS.
MSRDI, selects the defect for OTN layer as MSRDI.
MSREI, selects the defect for OTN layer as MSREI.
AUAIS, selects the defect for OTN layer as AUAIS.
AULOP, selects the defect for OTN layer as AULOP.
H4LOM, selects the defect for OTN layer as H4LOM.
HPRDI, selects the defect for OTN layer as HPRDI.
ESD, selects the defect for OTN layer as ESD.
ECD, selects the defect for OTN layer as ECD.
EPD, selects the defect for OTN layer as EPD.

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

HPREI, selects the defect for OTN layer as HPREI.

TUAIS, selects the defect for OTN layer as TUAIS.

TULOP, selects the defect for OTN layer as TULOP.

LPRFI, selects the defect for OTN layer as LPRFI.

LPRDI, selects the defect for OTN layer as LPRDI.

ESDLop, selects the defect for OTN layer as ESDLop.

ECDLop, selects the defect for OTN layer as ECDLop.

EPDLop, selects the defect for OTN layer as EPDLop.

LPREI, selects the defect for OTN layer as LPREI.

FCCW, selects the defect for OTN layer as FCCW.

FUCW, selects the defect for OTN layer as FUCW.

FCSymb, selects the defect for OTN layer as FCSymb.

FCBit, selects the defect for OTN layer as FCBit.

OAIS, selects the defect for OTN layer as OAIS.

OBDI, selects the defect for OTN layer as OBDI.

OLOF, selects the defect for OTN layer as OLOF.

OOOF, selects the defect for OTN layer as OOOF.

OLOM, selects the defect for OTN layer as OLOM.

OOM, selects the defect for OTN layer as OOM.

OBlae, selects the defect for OTN layer as OBlae.

OIAE, selects the defect for OTN layer as OIAE.

OBIP8, selects the defect for OTN layer as OBIP8.

OBEL, selects the defect for OTN layer as OBEL.

FAS, selects the defect for OTN layer as FAS.

MFAS, selects the defect for OTN layer as MFAS.

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

ODAIS, selects the defect for OTN layer as ODAIS.

ODBDI, selects the defect for OTN layer as ODBDI.

OLCK, selects the defect for OTN layer as OLCK.

OOCI, selects the defect for OTN layer as OOCI.

OFSF, selects the defect for OTN layer as OFSF.

OBSF, selects the defect for OTN layer as OBSF.

OFSD, selects the defect for OTN layer as OFSD.

OBSD selects the defect for OTN layer as OBSD.

LOFLom, selects the defect for OTN layer as LOFLom.

ODBIP8, selects the defect for OTN layer as ODBIP8.

ODBEI, selects the defect for OTN layer as ODBEI.

PLOsS, selects the defect for OTN layer as PLOsS.

BITerror, selects the defect for OTN layer as BITerror.

OPUCSF, selects the defect for OTN layer as OPUCSF.

Example(s)

```
SENS:DATA:TEL:SDT:OTN:LAY:TYPE SECT
SENS:DATA:TEL:SDT:OTN:DSEL BERR
SENS:DATA:TEL:SDT:OTN:DSEL? Returns
BERROR
```

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

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

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

Description	<p>This query returns the defect for specific layer for OTN.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:SENSE[1..n]:DATA:TELEcom:SDT:OTN: DSElection?</code>
Parameter(s)	None
Response Syntax	<Selection>
Response (s)	<p>Selection:</p> <p>The response data syntax for <Selection> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the defect for OTN layer.</p> <p>LOS, selects the defect for OTN layer as LOS.</p> <p>LOF, selects the defect for OTN layer as LOF.</p> <p>SEF, selects the defect for OTN layer as SEF.</p> <p>B1, selects the defect for OTN layer as B1.</p> <p>RFAS, selects the defect for OTN layer as RFAS.</p> <p>AISL, selects the defect for OTN layer as AISL.</p> <p>RDIL, selects the defect for OTN layer as RDIL.</p> <p>B2, selects the defect for OTN layer as B2.</p>

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

REIL, selects the defect for OTN layer as REIL.
AISP, selects the defect for OTN layer as AISP.
RDIP, selects the defect for OTN layer as RDIP.
EPSD, selects the defect for OTN layer as EPSD.
EPCD, selects the defect for OTN layer as EPCD.
EPPD, selects the defect for OTN layer as EPPD.
LOM, selects the defect for OTN layer as LOM.
LOPP, selects the defect for OTN layer as LOPP.
PDIP, selects the defect for OTN layer as PDIP.
B3, selects the defect for OTN layer as B3.
REIP, selects the defect for OTN layer as REIP.
AISV, selects the defect for OTN layer as AISV.
RDIV, selects the defect for OTN layer as RDIV.
EVSD, selects the defect for OTN layer as EVSD.
EVCD, selects the defect for OTN layer as EVCD.
EVPD, selects the defect for OTN layer as EVPD.
RFIV, selects the defect for OTN layer as RFIV.
LOPV, selects the defect for OTN layer as LOPV.
BIP2, selects the defect for OTN layer as BIP2.
REIV, selects the defect for OTN layer as REIV.
OOF, selects the defect for OTN layer as OOF.
MSAIS, selects the defect for OTN layer as MSAIS.
MSRDI, selects the defect for OTN layer as MSRDI.
MSREI, selects the defect for OTN layer as MSREI.
AUAIS, selects the defect for OTN layer as AUAIS.
AULOP, selects the defect for OTN layer as AULOP.
H4LOM, selects the defect for OTN layer as H4LOM.

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

HPRDI, selects the defect for OTN layer as HPRDI.
ESD, selects the defect for OTN layer as ESD.
ECD, selects the defect for OTN layer as ECD.
EPD, selects the defect for OTN layer as EPD.
HPREI, selects the defect for OTN layer as HPREI.
TUAIS, selects the defect for OTN layer as TUAIS.
TULOP, selects the defect for OTN layer as TULOP.
LPRFI, selects the defect for OTN layer as LPRFI.
LPRDI, selects the defect for OTN layer as LPRDI.
ESDLop, selects the defect for OTN layer as ESDLop.
ECDLop, selects the defect for OTN layer as ECDLop.
EPDLop, selects the defect for OTN layer as EPDLop.
LPREI, selects the defect for OTN layer as LPREI.
FCCW, selects the defect for OTN layer as FCCW.
FUCW, selects the defect for OTN layer as FUCW.
FCSYmb, selects the defect for OTN layer as FCSYmb.
FCBit, selects the defect for OTN layer as FCBit.
OAIS, selects the defect for OTN layer as OAIS.
OBDI, selects the defect for OTN layer as OBDI.
OLOF, selects the defect for OTN layer as OLOF.
OOOF, selects the defect for OTN layer as OOOF.
OLOM, selects the defect for OTN layer as OLOM.
OOM, selects the defect for OTN layer as OOM.
OBlae, selects the defect for OTN layer as OBlae.
OIAE, selects the defect for OTN layer as OIAE.

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

OBIP8, selects the defect for OTN layer as OBIP8.

OBEI, selects the defect for OTN layer as OBEI.

FAS, selects the defect for OTN layer as FAS.

MFAS, selects the defect for OTN layer as MFAS.

ODAIS, selects the defect for OTN layer as ODAIS.

ODBDI, selects the defect for OTN layer as ODBDI.

OLCK, selects the defect for OTN layer as OLCK.

OOCI, selects the defect for OTN layer as OOCI.

OFSF, selects the defect for OTN layer as OFSF.

OBSF, selects the defect for OTN layer as OBSF.

OFSD, selects the defect for OTN layer as OFSD.

OBSD selects the defect for OTN layer as OBSD.

LOFLom, selects the defect for OTN layer as LOFLom.

ODBIP8, selects the defect for OTN layer as ODBIP8.

ODBEI, selects the defect for OTN layer as ODBEI.

PLOSSs, selects the defect for OTN layer as PLOSSs.

BITerror, selects the defect for OTN layer as BITerror.

OPUCSF, selects the defect for OTN layer as OPUCSF.

Example(s)

```
SENS:DATA:TEL:SDT:OTN:LAY:TYPE SECT
SENS:DATA:TEL:SDT:OTN:DSEL BERR
SENS:DATA:TEL:SDT:OTN:DSEL? Returns
BERROR
```

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

See Also SENSe[1..n]:DATA:TELEcom:SDT:OTN:LAYer:TYPE
 SENSe[1..n]:DATA:TELEcom:SDT:OTN:DSElection
 SENSe[1..n]:DATA:TELEcom:SDT

EoOTN Command Reference

:SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:ADDRess:DESTination

Description

This command sets the Destination Media Access Control (MAC) address of the stream.

At *RST, this value is set to "FE:FE:FE:FE:FE:FE".

Syntax

:SOURce[1..n]:DATA:TELEcom:ETHernet:
STReam:ADDRess:DESTination <wsp> <Tgen>,
<Address>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
STReam:ADDRess:DESTination**

Parameter(s)	<p>Tgen: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the traffic stream.</p> <p>Address: The program data syntax for the second parameter is defined as a <STRING PROGRAM DATA> element. Sets the Destination Media Access Control (MAC) address.</p>
Example(s)	<p>* 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"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:ADDRess:DESTination?</p>

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

Description	<p>This query returns the Destination Media Access Control (MAC) address of the stream.</p> <p>At *RST, this value is set to "FE:FE:FE:FE:FE:FE".</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:ADDRes:DESTination?<wsp><Tgen></p>
Parameter(s)	<p>Tgen: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the traffic stream.</p>
Response Syntax	<p><Address></p>

:SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:ADDRes:DESTination?

Response(s)	Address: The response data syntax for <Address> is defined as a <STRING RESPONSE DATA> element. Returns the Destination Media Access Control (MAC) address in the form of a string.
Example(s)	* 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"
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:ADDRes:DESTination

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

This command sets the Source Media Access Control (MAC) address of the stream.

At *RST, this value is set to "00:00:00:00:00:00".

Syntax

:SOURCE[1..n]:DATA:TELEcom:ETHernet:
STReam:ADDReSS:SOURce <wsp> <Tgen>,
<Address>

:SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:ADDResS:SOURce

Parameter(s)	<p>Tgen:</p> <p>The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the traffic stream.</p> <p>Address:</p> <p>The program data syntax for the second parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the Source Media Access Control (MAC) address.</p>
Example(s)	<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 Returns "FE:FE:FE:FE:00:00"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:ADDResS:SOURce?</p>

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

Description	<p>This query returns the Source Media Access Control (MAC) address of the stream.</p> <p>At *RST, this value is set to "00:00:00:00:00:00".</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:ADDRess:SOURCE? <wsp> <Tgen></p>
Parameter(s)	<p>Tgen: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the traffic stream.</p>
Response Syntax	<p><Address></p>

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

Response(s)

Address:

The response data syntax for <Address> is defined as a <STRING RESPONSE DATA> element.

Returns the Source Media Access Control (MAC) address in the form of a string.

Example(s)

* SOUR:DATA:TEL:ETH:STR:ADDR:SOUR 1,
"FE:FE:FE:FE:00:00"

* SOUR:DATA:TEL:ETH:STR:ADDR:SOUR? 1
Returns "FE:FE:FE:FE:00:00"

See Also

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

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
STReam:DATAlink:SIZE**

Description	<p>This command sets the frame size for each traffic type.</p> <p>At *RST, this value is set to 64.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:DATAlink:SIZE<wsp><Tgen>,<Size> MAXimum MINimum</p>
Parameter(s)	<p>Tgen: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the stream from 1 to 10.</p>

:SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:DATAlink:SIZE

Parameter(s)	<p>Size:</p> <p>The program data syntax for the second parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the frame size for each traffic type.</p>
Example(s)	<p>* SOUR:DATA:TEL:ETH:STR:DAT:SIZE 1,550</p> <p>* SOUR:DATA:TEL:ETH:STR:DAT:SIZE? 1</p> <p>Returns 550</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:DATAlink</p> <p>* SOURce[1..n]:DATA:TELEcom:ETHernet STReam:DATAlink:SIZE?</p>

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

Description This query returns the frame size for each traffic type.

At *RST, this value is set to 64.

Syntax :SOURCE[1..n]:DATA:TELEcom:ETHernet:
STReam:DATAlink:SIZE? <wsp> <Tgen>
[,MAXimum|MINimum]

:SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:DATAlink:SIZE?

Parameter(s)

Tgen:

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the stream from 1 to 10.

The program data syntax for <Size> is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

MAXimum | MINimum.

MAXimum, retrieves the greatest supported value of the instrument.

MINimum, retrieves the smallest supported value of the instrument.

This parameter is optional. If no token is specified, the current datalink size will be returned.

Response Syntax

<Size>

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

Response(s)	<p>Size:</p> <p>The response data syntax for <Size> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the frame size for each traffic type.</p>
Example(s)	<p>* SOUR:DATA:TEL:ETH:STR:DAT:SIZE 1,550</p> <p>* SOUR:DATA:TEL:ETH:STR:DAT:SIZE? 1</p> <p>Returns 550</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:DATAlink</p> <p>* SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:DATAlink:SIZE</p>

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

Description This command selects the stream rate.

At *RST, this value is set to 100.0.

Syntax :SOURce[1..n]:DATA:TELEcom:ETHernet:
STReam:RATE<wsp> <Tgen>, <Rate>
|MAXimum | MINimum

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

Parameter(s)	<p>Tgen:</p> <p>The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the traffic stream</p> <p>Rate:</p> <p>The program data syntax for the second parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are:</p> <p>MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p> <p>Selects the stream rate.</p> <p>Choices are 0.001 through 100.00.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ETH:STR:RATE 1,100* SOUR:DATA:TEL:ETH:STR:RATE? 1 Returns 100
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:RATE?

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

Description	<p>This query returns the stream rate.</p> <p>At *RST, this value is set to 100.0.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:RATE? <wsp> <Tgen>, <Rate> [, MAXimum MINimum]</p>
Parameter(s)	<p>Tgen: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the traffic stream.</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum. MAXimum, retrieves the greatest supported value of the instrument. MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current stream rate will be returned.</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
STReam:RATE?****Response Syntax** <Rate>**Response(s)**

Rate:

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the stream rate.

Example(s)

* SOUR:DATA:TEL:ETH:STR:RATE 1,100

* SOUR:DATA:TEL:ETH:STR:RATE? 1 Returns 100

See Also

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

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

Description This command enables or disables Virtual Local Area Network (VLAN) stream.

At *RST, this value is set to OFF.

Syntax :SOURCE[1..n]:DATA:TELEcom:ETHernet:
STReam:VLAN<wsp> <Tgen>,<Set>

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

Parameter(s)	<p>Tgen:</p> <p>The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the traffic stream.</p> <p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Virtual Local Area Network (VLAN) stream.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ETH:STR:VLAN 1,ON* SOUR:DATA:TEL:ETH:STR:VLAN? 1 Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ETHernet:STReam:VLAN?

:SOURCE[1..n]:DATA:TELEcom:ETHernet: STream:VLAN?

Description	<p>This query returns the status of the Virtual Local Area Network (VLAN) stream.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet: STream:VLAN?<Tgen></p>
Parameter(s)	<p>Tgen: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the traffic stream.</p>
Response Syntax	<p><Set></p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
STReam:VLAN?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of the Virtual Local Area Network (VLAN) stream.

Example(s)

* SOUR:DATA:TEL:ETH:STR:VLAN 1,ON

* SOUR:DATA:TEL:ETH:STR:VLAN? 1 Returns 1

See Also

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

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

Description

This command sets the Virtual Local Area Network (VLAN) Identification (ID) of the stream.

At *RST, this value is set to 2.

Syntax

:SOURCE[1..n]:DATA:TELEcom:ETHernet:
STReam:VLAN:ID<wsp><Tgen>,<Stacked>,
<Vlanid> | MAXimum | MINimum

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
STream:VLAN:ID****Parameter(s)****Tgen:**

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the traffic stream.

Stacked:

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Select the Virtual Local Area Network (VLAN) stacked.

The value for stacked is set to 1 only.

Vlanid:

The program data syntax for the second parameter is defined as a <numeric_value> element.

The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum.

MAXimum allows to set the instrument to the greatest supported value.

MINimum allows to set the instrument to the smallest supported value.

Sets the Virtual Local Area Network (VLAN) ID of the stream.

Choices are 0 through 4095.

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

- Example(s)**
- * 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
- See Also**
- * 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:ID?****Description**

This query returns the Virtual Local Area Network (VLAN) Identification (ID) of the stream.

At *RST, this value is set to 2.

Syntax

:SOURCE[1..n]:DATA:TELEcom:ETHernet:
STReam:VLAN:ID?[<wsp> <Tgen>, <Stacked>
MAXimum | MINimum]

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

Parameter(s)

Tgen:

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the traffic stream.

Stacked:

The program data syntax for the second parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Select the Virtual Local Area Network (VLAN) stacked.

The value for stacked is set to 1 only.

The program data syntax for the third parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum | MINimum.

MAXimum, retrieves the greatest supported value of the instrument.

MINimum, retrieves the smallest supported value of the instrument.

This parameter is optional. If no token is specified, the current VLAN ID will be returned.

Response Syntax

<Vlanid>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
STReam:VLAN:ID?****Response(s)**

Vlanid:

The response data syntax for <Vlanid> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the Virtual Local Area Network (VLAN) ID of the stream.

Example(s)

* 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

See Also

* 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:PRiority

Description

This command sets the Virtual Local Area Network (VLAN) user priority.

At *RST, this value is set to 0.

Syntax

:SOURCE[1..n]:DATA:TELEcom:ETHernet:
STReam:VLAN:PRiority<wsp><Tgen>,
<Stacked>,<Priority>

:SOURce[1..n]:DATA:TELEcom:ETHernet: STream:VLAN:PRiority

Parameter(s)

Tgen:

The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the traffic stream.

Stacked:

The program data syntax for the second parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Select the Virtual Local Area Network (VLAN) stacked.

The value for stacked is set to 1 only.

Priority:

The program data syntax for the third parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the Virtual Local Area Network (VLAN) user priority.

Example(s)

* SOUR:DATA:TEL:ETH:STR:VLAN 1,ON

* SOUR:DATA:TEL:ETH:STR:VLAN:PRI 1,1,5

* SOUR:DATA:TEL:ETH:STR:VLAN:PRI? 1,1

Returns 5

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

See Also

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

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

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

Description	<p>This query returns the Virtual Local Area Network (VLAN) user priority.</p> <p>At *RST, this value is set to 0.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet: STReam:VLAN:PRiority?<wsp><Tgen>, <Stacked></p>
Parameter(s)	<p>Tgen:</p> <p>The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the traffic stream.</p> <p>Stacked:</p> <p>The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Select the Virtual Local Area Network (VLAN) stacked.</p> <p>The value for stacked is set to 1 only.</p>
Response Syntax	<p><Priority></p>

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

Response(s)

Priority:

The response data syntax for <Priority> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the Virtual Local Area Network (VLAN) user priority.

Example(s)

* SOUR:DATA:TEL:ETH:STR:VLAN 1,ON

* SOUR:DATA:TEL:ETH:STR:VLAN:PRI 1,1,5

* SOUR:DATA:TEL:ETH:STR:VLAN:PRI? 1,1

Returns 5

See Also

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

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

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

Description This query returns Virtual Local Area Network (VLAN) Ethernet type.

At *RST, this value is set to 8100.

Syntax :SOURCE[1..n]:DATA:TELEcom:ETHernet:
STReam:VLAN:TYPE?<wsp><Tgen>,
<Stacked>

Parameter(s) Tgen:
The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Selects the traffic stream.

Stacked:
The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Select the Virtual Local Area Network (VLAN) stacked.

The value for stacked is set to 1 only.

Response Syntax <Idtype>

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

Response(s)

Idtype:

The response data syntax for <Idtype> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Virtual Local Area Network (VLAN) Ethernet 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.

Example(s)

* SOUR:DATA:TEL:ETH:STR:VLAN 1,ON

* SOUR:DATA:TEL:ETH:STR:VLAN:TYPE? 1,1

Returns V8100

See Also

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

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

Description This command enables or disables the stream.

At *RST, this value is set to OFF.

Syntax :SOURCE[1..n]:DATA:TELEcom:ETHernet:
STReam:ENABled<wsp><Tgen>,<Set>

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

Parameter(s)	<p>Tgen: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the traffic stream.</p> <p>Set: The program data syntax for <Set> is defined as a <Boolean Program Data> element. The <Set> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0. Enables or disables the stream.</p>
Example(s)	<p>* SOUR:DATA:TEL:ETH:STR:ENAB 1,ON * SOUR:DATA:TEL:ETH:STR:ENAB? 1 Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:ENABled?</p>

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

Description	<p>This query returns the status of the stream.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:ETHernet: STream:ENABLEd? <Tgen></p>
Parameter(s)	<p>Tgen:</p> <p>The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Selects the traffic stream.</p>
Response Syntax	<p><Set></p>

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

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the stream.
Example(s)	* SOUR:DATA:TEL:ETH:STR:ENAB 1,ON * SOUR:DATA:TEL:ETH:STR:ENAB? 1 Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: STReam:ENABled

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical:HISTory?**

Description	<p>This query returns the history status of the physical alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:HISTory?<wsp>LDOWn LFAult RFAult</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LDOWn LFAult RFAult.</p> <p>Selects the physical alarm type.</p> <p>LDOWn, selects Link Down as physical alarm. LFAult, selects Local Fault as physical alarm. RFAult, selects Remote Fault as physical alarm.</p>
Response Syntax	<History>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical:HISTory?**

Response(s)

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the history status of physical 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)

* SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE LFA

* SOUR:DATA:TEL:ETH:ALAR:PHYS ON

* FETC:DATA:TEL:ETH:ALAR:PHYS:HIST? LFA

Returns the alarm history.

See Also

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical:TYPE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical:CURRent?**

Description	<p>This query returns the current status of the physical alarm.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:CURRent? <wsp>LDOWn LFAult RFAult</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LDOWn LFAult RFAult.</p> <p>Selects the physical alarm type.</p> <p>LDOWn, selects Link Down as physical alarm. LFAult, selects Local Fault as physical alarm. RFAult, selects Remote Fault as physical alarm.</p>
Response Syntax	<Current>

:FETCh[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:CURRent?

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current status of physical 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.

Example(s)

* SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE LFA

* SOUR:DATA:TEL:ETH:ALAR:PHYS ON

* FETC:DATA:TEL:ETH:ALAR:PHYS:CURR? LFA

Returns the current alarm status.

See Also

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical:TYPE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical:SEConds?**

Description	<p>This query returns the number of seconds within which physical alarm occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:SEConds?<wsp>LDOWn LFAult RFAult</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LDOWn LFAult RFAult.</p> <p>Selects the physical alarm type.</p> <p>LDOWn, selects Link Down as physical alarm. LFAult, selects Local Fault as physical alarm. RFAult, selects Remote Fault as physical alarm.</p>
Response Syntax	<Seconds>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of physical alarm.
Example(s)	* SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE LFA * SOUR:DATA:TEL:ETH:ALAR:PHYS ON * FETC:DATA:TEL:ETH:ALAR:PHYS:SEC? LFA Returns the number of seconds of physical alarm.
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:TYPE * SOURce[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:HISTory?**

Description	<p>This query returns the history status of the physical error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:HISTory? <wsp> BLOCk</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BLOCk.</p> <p>Selects the physical error type.</p> <p>BLOCk, selects Block as physical error.</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERROr:
PHYSical:HISTory?**

Response Syntax <History>

Response(s) History:
The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the history status of physical 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.

Example(s) * FETC:DATA:TEL:ETH:ERR:PHYS:HIST? BLOCK
Returns the error history.

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:CURRent?**

Description	<p>This query returns the current status of the physical error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:CURRent? <wsp>BLOCK</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BLOCK.</p> <p>Selects the physical error type.</p> <p>BLOCK, selects Block as physical error.</p>

:FETCh[1..n]:DATA:TELEcom:ETHernet:ERROr: PHYSical:CURRent?

Response Syntax <Current>

Response(s) Current:
The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the current status of physical 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) * FETC:DATA:TEL:ETH:ERR:PHYS:CURR? BLOCK
Returns the current error status.

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:SEConds?**

Description	<p>This query returns the number of seconds within which physical error occurred.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:SEConds? <wsp>BLOCK</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BLOCK.</p> <p>Selects the physical error type.</p> <p>BLOCK, selects Block as physical error.</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:SEConds?**

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of physical error.

Example(s) * FETC:DATA:TEL:ETH:ERR:PHYS:SEC? BLOCK
Returns the number of errored seconds.

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:COUnT?**

Description	<p>This query returns the count of the physical error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:COUnT? <wsp>BLOCk</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BLOCk.</p> <p>Selects the physical error type.</p> <p>BLOCk, selects Block as physical error.</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:COUNT?**

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of physical error.

Example(s) * FETC:DATA:TEL:ETH:ERR:PHYS:COUN? BLOCK
Returns the error count.

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:RATE?**

Description	<p>This query returns the current rate of the physical error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:RATE?<wsp>BLOCk</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BLOCk.</p> <p>Selects the physical error type.</p> <p>BLOCk, selects Block as physical error.</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:RATE?**

Response Syntax <Rate>

Response(s) Rate:
The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.
Returns the count of physical error.

Example(s) * FETC:DATA:TEL:ETH:ERR:PHYS:RATE? BLOCK
Returns the current error rate.

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:HISTory?****Description**

This query returns the history status of the Media Access Control (MAC) error.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:HISTory? <wsp>FCS | JABBer |
OVERsize | RUNT | UNDERsize | ALIGNment

:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:HISTory?

Parameter(s)	Error:
	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCS JABBer OVERsize RUNT UNDersize ALIGNment.</p> <p>Selects the Media Access Control (MAC) error type.</p> <p>FCS, selects Frame Check Sequence (FCS) as MAC error.</p> <p>JABBer, selects Jabber/Exhaustive as MAC error.</p> <p>OVERsize, selects Oversize as MAC error.</p> <p>RUNT, selects Runt as MAC error.</p> <p>UNDersize, selects Undersize as MAC error.</p> <p>ALIGNment, selects Alignment as MAC error.</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:HISTory?****Response Syntax** <History>**Response(s)**

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the history status of Media Access Control (MAC) 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.

Example(s)

* FETC:DATA:TEL:ETH:ERR:MAC:HIST? JABB

Returns the error history status.

:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:CURREnt?

Description This query returns the current status of the Media Access Control (MAC) error.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:CURREnt? <wsp>FCS|JABBer|
OVERsize|RUNT|UNDersize|ALIGNment

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:CURRent?**

Parameter(s)	Error:
	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCS JABBer OVERsize RUNT UNDersize ALIGNment.</p> <p>Selects the Media Access Control (MAC) error type.</p> <p>FCS, selects Frame Check Sequence (FCS) as MAC error.</p> <p>JABBer, selects Jabber/Exhaustive as MAC error.</p> <p>OVERsize, selects Oversize as MAC error.</p> <p>RUNT, selects Runt as MAC error.</p> <p>UNDersize, selects Undersize as MAC error.</p> <p>ALIGNment, selects Alignment as MAC error.</p>

:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:CURREnt?

Response Syntax <Current>

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current status of Media Access Control (MAC) 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)

* FETC:DATA:TEL:ETH:ERR:MAC:CURR? JABB
Returns the current error status.

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:SEConds?****Description**

This query returns the number of seconds within which Media Access Control (MAC) error occurred.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:SEConds? <wsp>FCS | JABBer |
OVERsize | RUNT | UNDERsize | ALIGNment

:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:SEConds?

Parameter(s)	Error:
	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCS JABBer OVERsize RUNT UNDersize ALIGNment.</p> <p>Selects the Media Access Control (MAC) error type.</p> <p>FCS, selects Frame Check Sequence (FCS) as MAC error.</p> <p>JABBer, selects Jabber/Exhaustive as MAC error.</p> <p>OVERsize, selects Oversize as MAC error.</p> <p>RUNT, selects Runt as MAC error.</p> <p>UNDersize, selects Undersize as MAC error.</p> <p>ALIGNment, selects Alignment as MAC error.</p>

**:FETCh[1..n]:DATA:TELecom:ETHernet:
ERRor:MAC:SEConds?****Response Syntax** <Seconds>**Response(s)**

Seconds:

The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the number of seconds within which Media Access Control (MAC) error occurred.

Example(s)

* FETC:DATA:TEL:ETH:ERR:MAC:SEC? JABB
Returns the number of errored seconds.

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:COUNT?**

Description

This query returns the count of the Media Access Control (MAC) error.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:COUNT?<wsp>FCS|JABBer|
OVERsize|RUNT|UNDersize|ALIGNment

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:COUNT?**

Parameter(s)	Error:
	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCS JABBer OVERsize RUNT UNDersize ALIGNment.</p> <p>Selects the Media Access Control (MAC) error type.</p> <p>FCS, selects Frame Check Sequence (FCS) as MAC error.</p> <p>JABBer, selects Jabber/Exhaustive as MAC error.</p> <p>OVERsize, selects Oversize as MAC error.</p> <p>RUNT, selects Runt as MAC error.</p> <p>UNDersize, selects Undersize as MAC error.</p> <p>ALIGNment, selects Alignment as MAC error.</p>

:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:COUNT?

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of Media Access Control (MAC) error.

Example(s) * FETC:DATA:TEL:ETH:ERR:MAC:COUN? JABB
Returns the error count.

:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:RATE?

Description	<p>This query returns the current rate of the Media Access Control (MAC) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:RATE?<wsp>FCS JABBer OVERsize RUNT UNDersize ALIGnment</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>FCS JABBer OVERsize RUNT UNDersize ALIGnment.</pre> <p>Selects the Media Access Control (MAC) error type.</p> <p>FCS, selects Frame Check Sequence (FCS) as MAC error.</p> <p>JABBer, selects Jabber/Exhaustive as MAC error.</p> <p>OVERsize, selects Oversize as MAC error.</p> <p>RUNT, selects Runt as MAC error.</p> <p>UNDersize, selects Undersize as MAC error.</p> <p>ALIGnment, selects Alignment as MAC error.</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:RATE?**

Response Syntax <Rate>

Response(s) Rate:
The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.
Returns the current rate of Media Access Control (MAC) error.

Example(s) * FETC:DATA:TEL:ETH:ERR:MAC:RATE? JABB
Returns the current error rate.

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical****Description**

This command enables and disables the physical alarm generation.

This command is set to OFF.

Syntax

:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm
:PHYSical <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical**

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the physical alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE LFA* SOUR:DATA:TEL:ETH:ALAR:PHYS ON* SOUR:DATA:TEL:ETH:ALAR:PHYS? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical:TYPE* SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical?

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical?**

Description	This query returns the status of the physical alarm generation. This command is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm:PHYSical?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of the physical alarm generation.
Example(s)	* SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE LFA * SOUR:DATA:TEL:ETH:ALAR:PHYS ON * SOUR:DATA:TEL:ETH:ALAR:PHYS? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:TYPE * SOURce[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical:TYPE**

Description	<p>This command selects the physical alarm type.</p> <p>At *RST, this value is set to LDOWn.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:ETHernet:ALARm :PHYSical:TYPE<wsp>LDOWn LFAult RFAult</pre>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LDOWn LFAult RFAult.</p> <p>Select the physical alarm type.</p> <p>LDOWn, selects Link Down as physical alarm.</p> <p>LFAult, selects Local Fault as physical alarm.</p> <p>RFAult, selects Remote Fault as physical alarm.</p>
Example(s)	<pre>* SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE LFA * SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE? Returns LFAULT</pre>
See Also	<pre>* SOURCE[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:TYPE?</pre>

:SOURce[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:TYPE?

Description	This query returns the physical alarm type. At *RST, this value is set to LDOWn.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ALARm: :PHYSical:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
ALARm:PHYSical:TYPE?**

Response(s)	Alarm: The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element. Returns the physical alarm type. LDOWN, Link Down is selected as physical alarm. LFAULT, Local Fault is selected as physical alarm. RFAULT, Remote Fault is selected as physical alarm.
Example(s)	* SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE LFA * SOUR:DATA:TEL:ETH:ALAR:PHYS:TYPE? Returns LFAULT
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: ALARm:PHYSical:TYPE

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:MANual:TYPE**

Description	<p>This command sets the manual type Media Access Control (MAC) error.</p> <p>At *RST, this value is set to FCS.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :MAC:MANual:TYPE<wsp>FCS</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: FCS.</p> <p>Selects the manual type Media Access Control (MAC) error.</p> <p>FCS, selects Frame Check Sequence (FCS) as MAC error.</p>
Example(s)	<p>* SOUR:DATA:TEL:ETH:ERR:MAC:MAN:TYPE FCS</p> <p>* SOUR:DATA:TEL:ETH:ERR:MAC:MAN:TYPE? Returns FCS</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:MANual:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:MANual:TYPE?**

Description	This query returns the manual type Media Access Control (MAC) error. At *RST, this value is set to FCS.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :MAC:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:MANual:TYPE?

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the manual type Media Access Control (MAC) error.</p> <p>FCS, Frame Check Sequence (FCS) is selected as MAC error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ETH:ERR:MAC:MAN:TYPE FCS* SOUR:DATA:TEL:ETH:ERR:MAC:MAN:TYPE? Returns FCS
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:AMOUNT****Description**

This command sets the amount of the Media Access Control (MAC) error to be injected into the instrument.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor
:MAC:AMOUNT<wsp> <Amount> | MAXimum |
MINimum

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:AMOUNT

Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum allows to set the instrument to the greatest supported value.</p> <p>MINimum allows to set the instrument to the smallest supported value.</p> <p>Sets the amount of Media Access Control (MAC) error.</p> <p>Choices are 1 through 50.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ETH:ERR:MAC:MAN:TYPE FCS* SOUR:DATA:TEL:ETH:ERR:MAC:AMO 50* SOUR:DATA:TEL:ETH:ERR:MAC:AMO? <p>Returns 50</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOUNT?

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:AMOut?**

Description	<p>This query returns the amount of the Media Access Control (MAC) error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :MAC:AMOut? [<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<pre><Amount></pre>

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:AMOut?

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Media Access Control (MAC) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ETH:ERR:MAC:MAN:TYPE FCS* SOUR:DATA:TEL:ETH:ERR:MAC:AMO 50* SOUR:DATA:TEL:ETH:ERR:MAC:AMO? <p>Returns 50</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AMOut

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:INJect**

Description	<p>This command injects the Media Access Control (MAC) error.</p> <p>This command is an event and is not associated with *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :MAC:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:ETH:ERR:MAC:MAN:TYPE FCS * SOUR:DATA:TEL:ETH:ERR:MAC:AMO 50 * SOUR:DATA:TEL:ETH:ERR:MAC:INJ
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:AMount

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:AUTomated:TYPE

Description	This command selects the automated Media Access Control (MAC) error. At *RST, this value is set to FCS.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :MAC:AUTomated:TYPE<wsp>FCS

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:AUTomated:TYPE

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: FCS.</p> <p>Selects the automated Media Access Control (MAC) error type.</p> <p>FCS, selects Frame Check Sequence (FCS) as automated MAC error.</p>
Example(s)	<p>* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE FCS</p> <p>* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE?</p> <p>Returns FCS</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:TYPE?</p> <p>* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:AUTomated:TYPE?**

Description	This query returns the automated Media Access Control (MAC) error. At *RST, this value is set to FCS.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :MAC:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:AUTomated:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the automated Media Access Control (MAC) error type.</p> <p>FCS, Frame Check Sequence (FCS) is selected as automated MAC error.</p>
Example(s)	<p>* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE FCS</p> <p>* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE?</p> <p>Returns FCS</p>
See Also	<p>* SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:TYPE</p> <p>* SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE</p> <p>* SOURCE[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated</p>

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:AUTomated:RATE

Description This command sets the automated rate for the Media Access Control (MAC) error.

At *RST, this value is set to 1.0E-04.

Syntax :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor
:MAC:AUTomated:RATE <wsp> <Rate>
|MAXimum|MINimum

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:AUTomated:RATE

Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the automated rate for the Media Access Control (MAC) error. Choices are 1.0E-09 through 1.0E-02.</p>
Example(s)	<pre>* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE FCS * SOUR:DATA:TEL:ETH:ERR:MAC:AUT:RATE 1.0E-09 * SOUR:DATA:TEL:ETH:ERR:MAC:AUT:RATE? Returns 1.0E-09</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:AUTomated:RATE? * SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:MAC:AUTomated</pre>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:AUTomated:RATE?**

Description	<p>This query returns the rate for the automated Media Access Control (MAC) error.</p> <p>At *RST, this value is set to 1.0E-04.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :MAC:AUTomated:RATE?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current injected rate will be returned.</p>
Response Syntax	<pre><Rate></pre>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:AUTomated:RATE?****Response(s)**

Rate:

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Example(s)

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE FCS

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:RATE

1.0E-09

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:RATE?

Returns 1.0E-09

See Also* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:AUTomated

Description	This command enables or disables the automated Media Access Control (MAC) error. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :MAC:AUTomated <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:AUTomated**

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the automated Media Access Control (MAC) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE FCS* SOUR:DATA:TEL:ETH:ERR:MAC:AUT ON* SOUR:DATA:TEL:ETH:ERR:MAC:AUT? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:MAC:AUTomated?

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:AUTomated?

Description	This query returns the status of the automated Media Access Control (MAC) error. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :MAC:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:AUTomated?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of the automated Media Access Control (MAC) error.

Example(s)

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE FCS

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT ON

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT? Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:AUTomated:CONTInuous

Description

This command enables or disables the automated Media Access Control (MAC) error injection continuously.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor
:MAC:AUTomated:CONTInuous <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:AUTomated:CONTInuous****Parameter(s)**

Set:

The program data syntax for <Set> is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability.

ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the automated Media Access Control (MAC) error injection.

Example(s)

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE FCS

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:CONT ON

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:CONT?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated:CONTInuous?

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: MAC:AUTomated:CONTInuous?

Description	This query returns the status of the automated Media Access Control (MAC) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :MAC:AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
MAC:AUTomated:CONTInuous?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of the automated Media Access Control (MAC) error injection.

Example(s)

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:TYPE FCS

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:CONT ON

* SOUR:DATA:TEL:ETH:ERR:MAC:AUT:CONT?

Returns 1

See Also* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated:TYPE* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:MAC:AUTomated:CONTInuous

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:MANual:TYPE**

Description	<p>This command sets manual type physical error.</p> <p>At *RST, this value is set to BLOCK.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :PHYSical:MANual:TYPE<wsp>BLOCK</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: BLOCK.</p> <p>Sets the manual type physical error.</p> <p>BLOCK, selects Block as manual type physical error.</p>
Example(s)	<p>* SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE BLOC</p> <p>* SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE? Returns BLOCK</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:MANual:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:MANual:TYPE?**

Description	This query returns the manual type physical error. At *RST, this value is set to BLOCK.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :PHYSical:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:MANual:TYPE?**

Response(s)	Error: The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the manual type physical error. BLOCK, Block is selected as manual type physical error.
Example(s)	* SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE BLOCK * SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE? Returns BLOCK
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AMOut**

Description This command sets the amount of physical error to be injected into the instrument.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor
:PHYSical:AMOut <wsp> <Amount>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AMOut**

Parameter(s)	Amount: The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the amount of physical error. Choices are 1 through 50.
Example(s)	* SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE BLOC * SOUR:DATA:TEL:ETH:ERR:PHYS:AMO 50 * SOUR:DATA:TEL:ETH:ERR:PHYS:AMO? Returns 50
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AMOut?

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AMOut?**

Description	<p>This query returns the amount of physical error injected into the instrument.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :PHYSical:AMOut?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<pre><Amount></pre>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AMOut?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of physical error.
Example(s)	* SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE BLOC * SOUR:DATA:TEL:ETH:ERR:PHYS:AMO 50 * SOUR:DATA:TEL:ETH:ERR:PHYS:AMO? Returns 50
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AMOut

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:INJect**

Description	<p>This command injects the physical error into the instrument.</p> <p>This command is an event and is not associated with *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :PHYSical:INJect
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:ETH:ERR:PHYS:MAN:TYPE BLOC * SOUR:DATA:TEL:ETH:ERR:PHYS:AMO 50 * SOUR:DATA:TEL:ETH:ERR:PHYS:INJ
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AMOUNT

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:AUTomated:TYPE

Description

This command selects the automated type physical error.

At *RST, this value is set to BLOCK.

Syntax

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor
:PHYSical:AUTomated:TYPE<wsp>BLOCK

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BLOCk.</p> <p>Selects the automated type physical error</p> <p>BLOCk, selects Block as automated type physical error.</p>
Example(s)	<p>* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE BLOC</p> <p>* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE? Returns BLOCK</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated:TYPE?</p> <p>* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated:TYPE?**

Description	This query returns the automated type physical error. At *RST, this value is set to BLOCK.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :PHYSical:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated:TYPE?****Response(s)**

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the automated type physical error.

BLOCK, Block is selected as automated type physical error.

Example(s)

* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE
BLOC

* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE?
Returns BLOCK

See Also

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:PHYSical:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:PHYSical:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:PHYSical:AUTomated

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:AUTomated:RATE

Description

This command sets the automated rate for the physical error.

At *RST, this value is set to 1.0E-04.

Syntax

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor
:PHYSical:AUTomated:RATE<wsp> <Rate>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated:RATE**

Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the automated rate for the physical error. Choices are 1.0E-09 through 1.0E-02.</p>
Example(s)	<pre>* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE BLOC * SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:RATE 1.0E-09 * SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:RATE? Returns 1.0E-09</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated:RATE? * SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated</pre>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated:RATE?**

Description	<p>This query returns the automated rate for physical error.</p> <p>At *RST, this value is set to 1.0E-04.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :PHYSical:AUTomated:RATE? [<wsp> MAXimum MINimum]</p>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current physical rate will be returned.</p>
Response Syntax	<p><Rate></p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated:RATE?****Response(s)**

Rate:

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the automated rate for physical error.

Example(s)

* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE
BLOC

* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:RATE
1.0E-09

* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:RATE?
Returns 1.0E-09

See Also

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:PHYSical:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:PHYSical:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:PHYSical:AUTomated

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:AUTomated

Description

This command enables or disables the automated type physical error.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor
:PHYSical:AUTomated <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated**

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the automated type physical error.</p>
Example(s)	<p>* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE BLOC</p> <p>* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT ON</p> <p>* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT?</p> <p>Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated?</p>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated?**

Description	This query returns the status of automated type physical error. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :PHYSical:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of automated type physical error.

Example(s)

* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE
BLOC

* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT ON

* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:PHYSical:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:PHYSical:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:ETHernet:
ERRor:PHYSical:AUTomated

:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor: PHYSical:AUTomated:CONTInuous

Description

This command enables or disables the automated type physical error injection continuously.

At *RST, this value is set to OFF.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor  
:PHYSical:AUTomated:CONTInuous <wsp>  
<Set>
```

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated:CONTInuous****Parameter(s)**

Set:

The program data syntax for <Set> is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability.

ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the automated type physical error injection continuously.

Example(s)

```
* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE  
BLOC
```

```
* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:CONT  
ON
```

```
* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:CONT?  
Returns 1
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:ETHernet:  
ERRor:PHYSical:AUTomated:TYPE
```

```
* SOURce[1..n]:DATA:TELEcom:ETHernet:  
ERRor:PHYSical:AUTomated
```

```
* SOURce[1..n]:DATA:TELEcom:ETHernet:  
ERRor:PHYSical:AUTomated:CONTInuous?
```

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated:CONTInuous?**

Description	This query returns the status of automated type physical error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor :PHYSical:AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:ERRor:
PHYSical:AUTomated:CONTInuous?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of automated type physical error injection.
Example(s)	* SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:TYPE BLOC * SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:CONT ON * SOUR:DATA:TEL:ETH:ERR:PHYS:AUT:CONT? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated * SOURce[1..n]:DATA:TELEcom:ETHernet: ERRor:PHYSical:AUTomated:CONTInuous

:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe: BANDwidth?

Description	<p>This query returns the receiving data rate expressed in Mbps.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe: BANDwidth?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Bandwidth></p>
Response(s)	<p>Bandwidth: The response data syntax for <Bandwidth> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the receiving data rate.</p>
Example(s)	<p>* SENS:DATA:TEL:ETH:FRAM:BAND? Returns the receiving data rate.</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAME:
COUNT:RX?**

Description	<p>This query returns the valid frame count for the receiver.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAME: COUNT:RX? <wsp>MULTicast BROadcast UNICast NUNicast FTotal</pre>
Parameter(s)	<p>Ftype:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MULTicast BROadcast UNICast NUNicast FTotal.</p> <p>Selects the valid frame count for the receiver.</p> <p>MULTicast, selects Multicast as valid frame count.</p> <p>BROadcast, selects Broadcast as valid frame count.</p> <p>UNICast, selects Unicast as valid frame count.</p> <p>NUNicast, selects Nunicast as valid frame count.</p> <p>FTotal, selects Total as valid frame count.</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe:
COUNT:RX?**

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.

Returns the valid frame count for the receiver.

Example(s) * SENS:DATA:TEL:ETH:FRAM:COUN:RX? MULT
Returns the valid frame count for the receiver.

**:SOURce[1..n]:DATA:TELEcom:ETHernet:FRAME
:COUNT:TX?**

Description	<p>This query returns the valid frame count for the transmitter.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:ETHernet:FRAME :COUNT:TX?<wsp>MULTicast BROadcast UNICast NUNicast FTOtal</pre>
Parameter(s)	<p>Ftype:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MULTicast BROadcast UNICast NUNicast FTOtal.</p> <p>Selects the valid frame count for the transmitter.</p> <p>MULTicast, selects Multicast as valid frame count.</p> <p>BROadcast, selects Broadcast valid frame count.</p> <p>UNICast, selects Unicast valid frame count.</p> <p>NUNicast, selects Nunicast valid frame count.</p> <p>FTOtal, selects Total valid frame count.</p>

:SOURce[1..n]:DATA:TELEcom:ETHernet:FRAME :COUNT:TX?

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.

Returns the valid frame count for the transmitter.

Example(s) * SOUR:DATA:TEL:ETH:FRAM:COUN:TX? MULT
Returns the valid frame count for the transmitter.

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe:
RATE?**

Description	<p>This query returns the frame rate.</p> <p>At *RST, this value is device dependent.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe: RATE?
Parameter(s)	None
Response Syntax	<Rate>
Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the frame rate.</p>
Example(s)	<p>* SENS:DATA:TEL:ETH:FRAM:RATE?</p> <p>Returns the frame rate.</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe:
UTILization?**

Description	<p>This query returns the percentage of line rate utilization.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:ETHernet:FRAMe: UTILization?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Utilization></p>
Response(s)	<p>Utilization:</p> <p>The response data syntax for <Utilization> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the percentage of line rate utilization.</p>
Example(s)	<p>* SENS:DATA:TEL:ETH:FRAM:UTIL?</p> <p>Returns the rate utilization.</p>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZE:
COUNT?**

Description	<p>This query returns count of each received frame size (valid and invalid).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZE: COUNT?<wsp>FLESS64 F64 F65TO127 F128TO255 F256TO511 F512TO1023 F1024TO1518 FMORE1519 FSTotal</pre>
Parameter(s)	<p>Ftype:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>FLESS64 F64 F65TO127 F128TO255 F256TO511 F512TO1023 F1024TO1518 FMORE1519 FSTotal.</pre>

:SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZE:COUNT?

Selects the frame size.

FLESS64, selects the frame size less than 64.

F64, selects the frame size equal to 64.

F65TO127, selects the frame size between 65 to 127.

F128TO255, selects the frame size between 128 to 255.

F256TO511, selects the frame size between 256 to 511.

F512TO1023, selects the frame size between 512 to 1023.

F1024TO1518, selects the frame size between 1024 to 1518.

FMORE1519, selects the frame size more than 1519.

FSTotal, selects the total frame size.

**:SENSe[1..n]:DATA:TELEcom:ETHernet:FSIZe:
COUNT?****Response Syntax** <Count>**Response(s)** Count:
The response data syntax for <Count> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the count of each received frame size.**Example(s)** * SENS:DATA:TEL:ETH:FSIZ:COUN? F64
Returns the count of frame size.

Summary Page Command Reference

:FETCh[1..n]:DATA:TELEcom:SUMMery:TEST: HISTory?

Description

This query returns the summary of the history status of any alarms/errors related to the tests such as Port, OTN, SONET/SDH, DSn/PDH, Pattern, and Other. It also returns the status of Log Full, if exceeds its maximum capacity of 5000 events.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:SUMMery:TEST:
HISTory? <wsp>GLOBal|LFULl

**:FETCh[1..n]:DATA:TELEcom:SUMMArY:TEST:
HISTOrY?****Parameter(s)**

Type:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: GLOBal|LFULl.

Selects the test parameters for history status.

GLOBal, selects Global, which indicates the presence of any alarms/errors related to the tests such as Port, OTN, SONET/SDH, DSn/PDH, Pattern, and Other.

LFULl, selects Log Full (LFULl), which indicates that the logger exceeds its maximum capacity of 5000 events.

Response Syntax

<History>

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

Response(s)

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the summary of the history status of any alarms/errors related to the test.

PRESENT, indicates that at least one alarm/error has occurred.

ABSENT, indicates that no alarm/error occurred.

INACTIVE, indicates that the test did not run yet.

Example(s)

* FETC:DATA:TEL:SUMM:TEST:HIST? GLOB

Returns the summary of the history status of any alarms/errors related to Global.

See Also

* FETCh[1..n]:DATA:TELEcom:SUMMery:ALARm:
TEST:CURRent?

**:FETCh[1..n]:DATA:TELEcom:SUMMary:TEST:
CURRent?****Description**

This query returns the summary of the current status of any alarms/errors related to the tests such as Port, OTN, SONET/SDH, DS_n/PDH, Pattern, and Other. It also returns the status of Log Full, if logger exceeds its maximum capacity of 5000 events.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:SUMMary:TEST:
CURRent?<wsp>GLOBal|LFULl

Parameter(s)

Type:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: GLOBal|LFULl.

Selects the test parameters for history status.

GLOBal, selects Global, which indicates the presence of any alarms/errors related to the tests such as Port, OTN, SONET/SDH, DS_n/PDH, Pattern, and Other.

LFULl, selects Log Full (LFULl), which indicates that the logger exceeds its maximum capacity of 5000 events.

:FETCh[1..n]:DATA:TELEcom:SUMMary:TEST: CURRENT?

Response Syntax <Current>

Response(s) Current:
The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the summary of the current status of any alarms/errors related to the test.
PRESENT, indicates that at least one alarm/error has occurred in the last second.
ABSENT, indicates that there is no alarm/error.
INACTIVE, indicates that the test is not running.

Example(s) * FETC:DATA:TEL:SUMM:TEST:CURR? GLOB
Returns the summary of the current status of any alarms/errors related to Global.

See Also * FETCh[1..n]:DATA:TELEcom:SUMMary:TEST:
HISTory?

**:FETCh[1..n]:DATA:TELEcom:SUMMery:PORT:
HISTory?**

Description	<p>This query returns the summary of the history status of any alarms/errors related to the physical port such as LOS, Frequency, LOC, and Errors.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SUMMery:PORT: HISTory? <wsp>LOS FREQuency ERRors LOC</p>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOS FREQuency ERRors LOC.</p> <p>Selects the physical port type.</p> <p>LOS, selects the alarms/errors related to Loss of Signal (LOS).</p> <p>FREQuency, selects the alarms/errors related to Frequency.</p> <p>Errors, selects the Errors.</p> <p>LOC, selects the alarms/errors related to Loss of Clock (LOC).</p>

**:FETCh[1..n]:DATA:TELEcom:SUMMery:PORT:
HISTory?**

Response Syntax <History>

Response(s)

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the summary of the history status of any alarms/errors related to Port.

PRESENT, indicates that at least one alarm/error has occurred.

ABSENT, indicates that no alarm/error occurred.

INACTIVE, indicates that the test did not run yet.

Example(s)

* FETC:DATA:TEL:SUMM:PORT:HIST? LOS

Returns the summary of the history status of any alarms/errors related to LOS.

See Also

* FETCh[1..n]:DATA:TELEcom:SUMMery:PORT:
CURRent?

**:FETCh[1..n]:DATA:TELEcom:SUMMery:PORT:
CURRent?**

Description	<p>This query returns the summary of the current status of any alarms/errors related to the physical port such as LOS, Frequency, LOC, and Errors.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SUMMery:PORT: CURRent? <wsp>LOS FREQUency ERRors LOC</p>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOS FREQUency ERRors LOC.</p> <p>Selects the physical port type.</p> <p>LOS, selects the alarms/errors related to Loss of Signal (LOS).</p> <p>FREQUency, selects the alarms/errors related to Frequency.</p> <p>Errors, selects the Errors.</p> <p>LOC, selects the alarms/errors related to Loss of Clock (LOC).</p>

:FETCh[1..n]:DATA:TELEcom:SUMMary:PORT: CURRENT?

Response Syntax <Current>

Response(s) Current:
The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the summary of the current status of any alarms/errors related to Port.
PRESENT, indicates that at least one alarm/error has occurred in the last second.
ABSENT, indicates that there is no alarm/error.
INACTIVE, indicates that the test is not running.

Example(s) * FETC:DATA:TEL:SUMM:PORT:CURR? LOS
Returns the summary of the current status of any alarms/errors related to LOS.

See Also * FETCh[1..n]:DATA:TELEcom:SUMMary:PORT:
HISTory?

**:FETCh[1..n]:DATA:TELEcom:SUMMery:OTN:
HISTory?**

Description	<p>This query returns the summary of the history status of any alarms/errors related to the OTN such as OTU, ODU (includes ODU TCM alarms), and OPU.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SUMMery:OTN: HISTory? <wsp>OTU ODU OPU</pre>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>OTU ODU OPU.</p> <p>Selects the OTN type alarms/errors.</p> <p>OTU, selects Optical Transport Unit (OTU) as OTN type.</p> <p>ODU, selects Optical Data Unit (ODU) as OTN type.</p> <p>OPU, selects Optical Payload Unit (OPU) as OTN type.</p>

**:FETCh[1..n]:DATA:TELEcom:SUMMery:OTN:
HISTory?**

Response Syntax <History>

Response(s)

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the summary of the history status of any alarms/errors related to OTN.

PRESENT, indicates that at least one alarm/error has occurred.

ABSENT, indicates that no alarm/error occurred.

INACTIVE, indicates that the test did not run yet.

Example(s)

* FETC:DATA:TEL:SUMM:OTN:HIST? ODU

Returns the summary of the history status of any alarms/errors related to ODU.

See Also

* FETCh[1..n]:DATA:TELEcom:SUMMery:OTN:
CURRent?

**:FETCh[1..n]:DATA:TELEcom:SUMMARY:OTN:
CURRent?****Description**

This query returns the summary of the current status of any alarms/errors related to the OTN such as OTU, ODU (includes ODU TCM alarms), and OPU.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:SUMMARY:OTN:
CURRent? <wsp>OTU|ODU|OPU

Parameter(s)

Type:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

OTU|ODU|OPU.

Selects the OTN type alarms/errors.

OTU, selects Optical Transport Unit (OTU) as OTN type.

ODU, selects Optical Data Unit (ODU) as OTN type.

OPU, selects Optical Payload Unit (OPU) as OTN type.

**:FETCh[1..n]:DATA:TELEcom:SUMMery:OTN:
CURRENT?**

Response Syntax <Current>

Response(s) Current:
The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the summary of the current status of any alarms/errors related to OTN.
PRESENT, indicates that at least one alarm/error has occurred in the last second.
ABSENT, indicates that there is no alarm/error.
INACTIVE, indicates that the test is not running.

Example(s) * FETC:DATA:TEL:SUMM:OTN:CURR? ODU
Returns the summary of the current status of any alarms/errors related to ODU.

See Also * FETCh[1..n]:DATA:TELEcom:SUMMery:OTN:
HISTory?

**:FETCh[1..n]:DATA:TELEcom:SUMMery:
SONetsdh:HISTory?****Description**

This query returns the summary of the history status of any alarms/errors related to SONET/SDH testing such as Section/RS, Line/MS, High Order Path (HOP), and Low Order Path (LOP).

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:SUMMery:
SONetsdh:HISTory? <wsp>SECTionrs | LINems |
HOP | LOP

**:FETCh[1..n]:DATA:TELEcom:SUMMArY:
SONetsdh:HISTOrY?**

Parameter(s)	Type: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SECTIonrs LINems HOP LOP. Selects the SONET/SDH testing type. SECTIonrs, selects Section/RS as SONET/SDH testing type. LINems, selects Line/MS as SONET/SDH testing type. HOP, selects High Order Path (HOP) as SONET/SDH testing type. LOP, selects Low Order Path (LOP) as SONET/SDH testing type.
---------------------	--

Response Syntax	<History>
------------------------	-----------

**:FETCh[1..n]:DATA:TELEcom:SUMMary:
SONetsdh:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the summary of the history status of any alarms/errors related to SONET/SDH.</p> <p>PRESENT, indicates that at least one alarm/error has occurred.</p> <p>ABSENT, indicates that no alarm/error occurred.</p> <p>INACTIVE, indicates that the test did not run yet.</p>
Example(s)	<p>* FETC:DATA:TEL:SUMM:SON:HIST? HOP</p> <p>Returns the summary of the history status of any alarms/errors related to HOP.</p>
See Also	<p>* FETCh[1..n]:DATA:TELEcom:SUMMary: SONetsdh:CURRent?</p>

:FETCh[1..n]:DATA:TELEcom:SUMMery: SONetsdh:CURRent?

Description

This query returns the summary of the current status of any alarms/errors related to SONET/SDH testing such as Section/RS, Line/MS, High Order Path (HOP), and Low Order Path (LOP).

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:SUMMery:  
SONetsdh:CURRent?<wsp>SECTIonrs | LINems |  
HOP | LOP
```


**:FETCh[1..n]:DATA:TELEcom:SUMMArY:
SONetsdh:CURREnt?**

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SECTIonrs LINems HOP LOP.</p> <p>Selects the SONET/SDH testing type.</p> <p>SECTIonrs, selects Section/RS as SONET/SDH testing type.</p> <p>LINems, selects Line/MS as SONET/SDH testing type.</p> <p>HOP, selects High Order Path (HOP) as SONET/SDH testing type.</p> <p>LOP, selects Low Order Path (LOP) as SONET/SDH testing type.</p>
Response Syntax	<Current>

:FETCh[1..n]:DATA:TELEcom:SUMMery: SONetsdh:CURRent?

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the summary of the current status of any alarms/errors related to SONET/SDH.

PRESENT, indicates that at least one alarm/error has occurred in the last second.

ABSENT, indicates that there is no alarm/error.

INACTIVE, indicates that the test is not running.

Example(s)

* FETC:DATA:TEL:SUMM:SON:CURR? HOP

Returns the summary of the current status of any alarms/errors related to HOP.

See Also

* FETCh[1..n]:DATA:TELEcom:SUMMery:
SONetsdh:HISTory?

:FETCh[1..n]:DATA:TELEcom:SUMMARY:DSNPdh:HISTory?

Description	<p>This query returns the summary of the history status of any alarms/errors related to DS_n/PDH testing such as DS1/1.5M, DS3/45M, E1/2M, E2/8M, E3/34M, and E4/140M.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:SUMMARY:DSNPdh: HISTory? <wsp>DS1 DS3 E1 E2 E3 E4</pre>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DS1 DS3 E1 E2 E3 E4.</p> <p>Selects the DS_n/PDH testing type.</p> <p>DS1, selects DS1 as DS_n/PDH testing type.</p> <p>DS3, selects DS3 as DS_n/PDH testing type.</p> <p>E1, selects E1 as DS_n/PDH testing type.</p> <p>E2, selects E2 as DS_n/PDH testing type.</p> <p>E3, selects E3 as DS_n/PDH testing type.</p> <p>E4, selects E4 as DS_n/PDH testing type.</p>

**:FETCh[1..n]:DATA:TELEcom:SUMMery:
DSNPdh:HISTory?**

Response Syntax <History>

Response(s)

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the summary of the history status of any alarms/errors related to DS_n/PDH.

PRESENT, indicates that at least one alarm/error has occurred.

ABSENT, indicates that no alarm/error occurred.

INACTIVE, indicates that the test did not run yet.

Example(s)

* FETC:DATA:TEL:SUMM:DSNP:HIST? DS1

Returns the summary of the history status of any alarms/errors related to DS1.

See Also

* FETCh[1..n]:DATA:TELEcom:SUMMery:
DSNPdh:CURRent?

:FETCh[1..n]:DATA:TELEcom:SUMMARY:DSNPdh:CURRent?**Description**

This query returns the summary of the current status of any alarms/errors related to DS_n/PDH testing such as DS1/1.5M, DS3/45M, E1/2M, E2/8M, E3/34M, and E4/140M.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:SUMMARY:DSNPdh:CURRent? <wsp>DS1 | DS3 | E1 | E2 | E3 | E4

Parameter(s)

Type:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

DS1 | DS3 | E1 | E2 | E3 | E4.

Selects the DS_n/PDH testing type.

DS1, selects DS1 as DS_n/PDH testing type.

DS3, selects DS3 as DS_n/PDH testing type.

E1, selects E1 as DS_n/PDH testing type.

E2, selects E2 as DS_n/PDH testing type.

E3, selects E3 as DS_n/PDH testing type.

E4, selects E4 as DS_n/PDH testing type.

**:FETCh[1..n]:DATA:TELEcom:SUMMary:
DSNPdh:CURRent?**

Response Syntax <Current>

Response(s) Current:
The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.
Returns the summary of the current status of any alarms/errors related to DSn/PDH.
PRESENT, indicates that at least one alarm/error has occurred in the last second.
ABSENT, indicates that there is no alarm/error.
INACTIVE, indicates that the test is not running.

Example(s) * FETC:DATA:TEL:SUMM:DSNP:CURR? DS1
Returns the summary of the current status of any alarms/errors related to DS1.

See Also * FETCh[1..n]:DATA:TELEcom:SUMMary:
DSNPdh:HISTory?

**:FETCh[1..n]:DATA:TELEcom:SUMMery:
PATtern:HISTory?**

Description	<p>This query returns the summary of the history status of any alarms/errors related to pattern testing such as Bit Error and Pattern Loss.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SUMMery:PATtern: HISTory? <wsp>PLOSs BIT</p>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PLOSs BIT.</p> <p>Selects the Pattern testing type.</p> <p>PLOSs, selects Pattern Loss (PLOSs) as Pattern testing type.</p> <p>BIT, selects Bit Error (Bit) as Pattern testing type.</p>
Response Syntax	<p><History></p>

**:FETCh[1..n]:DATA:TELEcom:SUMMery:
PATTErn:HISTory?**

Response(s)

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the summary of the history status of any alarms/errors related to Pattern.

PRESENT, indicates that at least one alarm/error has occurred.

ABSENT, indicates that no alarm/error occurred.

INACTIVE, indicates that the test did not run yet.

Example(s)

* FETC:DATA:TEL:SUMM:PATT:HIST? PLOS

Returns the summary of the history status of any alarms/errors related to Pattern Loss.

See Also

* FETCh[1..n]:DATA:TELEcom:SUMMery:
PATTErn:CURREnt?

**:FETCh[1..n]:DATA:TELEcom:SUMMARY:
PATtern:CURRent?**

Description	<p>This query returns the summary of the current status of any alarms/errors related to pattern testing such as Bit Error and Pattern Loss.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SUMMARY:PATtern: CURRent? <wsp>PLOSs BIT</p>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PLOSs BIT.</p> <p>Selects the Pattern testing type.</p> <p>PLOSs, selects Pattern Loss (PLOSs) as Pattern testing type.</p> <p>Bit, selects Bit Error (Bit) as Pattern testing type.</p>
Response Syntax	<p><Current></p>

:FETCh[1..n]:DATA:TELEcom:SUMMery: PATtern:CURRent?

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the summary of the current status of any alarms/errors related to Pattern.

PRESENT, indicates that at least one alarm/error has occurred in the last second.

ABSENT, indicates that there is no alarm/error.

INACTIVE, indicates that the test is not running.

Example(s)

* FETC:DATA:TEL:SUMM:PATT:CURR? PLOS

Returns the summary of the current status of any alarms/errors related to Pattern Loss.

See Also

* FETCh[1..n]:DATA:TELEcom:SUMMery:
PATtern:HISTory?

**:FETCh[1..n]:DATA:TELEcom:SUMMery:OTHer:
HISTory?**

Description	<p>This query returns the summary of the history status of all other alarms/errors such as SDT.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SUMMery:OTHer: HISTory? <wsp>SDT</p>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: SDT.</p> <p>Selects Other alarm/error type.</p> <p>SDT, selects Service Disruption Time (SDT) as alarm/error.</p>
Response Syntax	<p><History></p>

**:FETCh[1..n]:DATA:TELEcom:SUMMery:OTHer:
HISTory?**

Response(s)

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the summary of the history status of any alarms/errors related to other test.

PRESENT, indicates that at least one alarm/error has occurred.

ABSENT, indicates that no alarm/error occurred.

INACTIVE, indicates that the test did not run yet.

Example(s)

* FETC:DATA:TEL:SUMM:OTH:HIST? SDT

Returns the summary of the history status of any alarms/errors related to SDT.

See Also

* FETCh[1..n]:DATA:TELEcom:SUMMery:OTHer:
CURRent?

**:FETCh[1..n]:DATA:TELEcom:SUMMery:OTHer:
CURRent?**

Description	<p>This query returns the summary of the current status of all other alarms/errors such as SDT.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SUMMery:OTHer: CURRent?<wsp>SDT</p>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: SDT.</p> <p>Selects Other alarm/error type.</p> <p>SDT, selects Service Disruption Time (SDT) as alarm/error.</p>
Response Syntax	<p><Current></p>

**:FETCh[1..n]:DATA:TELEcom:SUMMery:OTHer:
CURRENT?**

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the summary of the current status of any alarms/errors related to other test.

PRESENT, indicates that at least one alarm/error has occurred in the last second.

ABSENT, indicates that there is no alarm/error.

INACTIVE, indicates that the test is not running.

Example(s)

* FETC:DATA:TEL:SUMM:OTH:CURR? SDT

Returns the summary of the current status of any alarms/errors related to SDT.

See Also

* FETCh[1..n]:DATA:TELEcom:SUMMery:OTHer:
HISTory?

**:FETCh[1..n]:DATA:TELEcom:SUMMery:
COFFset:HISTory?****Description**

This query returns the summary of the history status of client offset.

At *RST, this value is device dependent.

Syntax

FETCh[1..n]:DATA:TELEcom:SUMMery:COFFset:
HISTory? <wsp> |FREQuency|

**:FETCh[1..n]:DATA:TELEcom:SUMMArY:
COFFset:HISTOrY?**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter is: FREQUency.</p> <p>Sets the frequency history status of client offset.</p>
Response Syntax	<p><History></p>
Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <Character response data> element.</p> <p>Returns the the summary of the history status of client offset.</p> <p>PRESENT, indicates that at least one alarm/error has occurred.</p> <p>ABSENT, indicates that no alarm/error occurred.</p>

**:FETCh[1..n]:DATA:TELEcom:SUMMery:
COFFset:CURRent?****Description**

This query returns the the summary of the current status of client offset.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:SUMMery:COFFset:
CURRent? <wsp> |FREQuency|

**:FETCh[1..n]:DATA:TELEcom:SUMMery:
COFFset:CURRent?**

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: FREQuency.</p> <p>Sets the current frequency status of client offset.</p>
Response Syntax	<Current>
Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <Character response data> element.</p> <p>Returns the the summary of the current status of client offset.</p> <p>PRESENT, indicates that at least one alarm/error has occurred.</p> <p>ABSENT, indicates that no alarm/error occurred.</p>

**:FETCh[1..n]:DATA:TELEcom:SUMMery:
COFFset:FREQUency?**

Description	<p>This query returns the frequency of the input signal in bps.</p> <p>At *RST, this value is set to disabled.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SUMMery:COFFset:FREQUency?
Parameter(s)	None
Response Syntax	<Frequency>
Response(s)	<p>Frequency:</p> <p>The response data syntax for <Frequency> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the frequency of the input signal in bps.</p>

**:FETCh[1..n]:DATA:TELEcom:SUMMery:
COFFset:OFFSet?**

Description	This query returns the frequency offset unit. At *RST, this value is set to ppm.
Syntax	:FETCh[1..n]:DATA:TELEcom:SUMMery:COFFset: OFFSet?
Parameter(s)	None
Response Syntax	<Offset>
Response(s)	Offset: The response data syntax for <Offset> is defined as a <STRING RESPONSE DATA> element. Returns the frequency offset unit.

Exhaustive Command Reference

:FETCh[1..n]:DATA:TELecom:ALARm:HISTory?

Description

This query returns the history status of all alarms related to the tests such as Port, Pattern, SDT/RTD, OTN, SONET, SDH, DS_n, and PDH. It also returns the combined status of all the tests.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELecom:ALARm:HISTory?  
<wsp>SECTion|LINE|HOP|PATtern|PORT|E1|  
E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|OTU2|  
OTU3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUT  
CM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|OD  
UTCM3|ODU1E|ODU2E|OPU1|OPU2|OPU3|OP  
U1E|OPU2E|SDT|GFP|ETH|OTU1F|OTU2F|OD  
U1F|ODU2F|ODUTCM1F|ODUTCM2F|OPU1F|O  
PU2F|ALL
```

:FETCh[1..n]:DATA:TELecom:ALARm:HISTory?

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

SECTion|LINE|HOP|PATTern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|OTU2|OTU3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|OPU1|OPU2|OPU3|OPU1E|OPU2E|SDT|GFP|ETH|OTU1F|OTU2F|ODU1F|ODU2F|ODUTCM1F|ODUTCM2F|OPU1F|OPU2F|ALL.

Selects all alarms related to the selected test or all the tests.

SECTion, retrieves all alarms related to Section.

LINE, retrieves all alarms related to Line.

HOP, retrieves all alarms related to HOP.

PATTern, retrieves all alarms related to Pattern.

PORT, retrieves all alarms related to Port.

E1, retrieves all alarms related to E1.

E2, retrieves all alarms related to E2.

E3, retrieves all alarms related to E3.

E4, retrieves all alarms related to E4.

RS, retrieves all alarms related to RS.

MS, retrieves all alarms related to MS.

LOP, retrieves all alarms related to LOP.

DS1, retrieves all alarms related to DS1.

:FETCh[1..n]:DATA:TELecom:ALARm:HISTory?

DS3, retrieves all alarms related to DS3.

OTU1, retrieves all alarms related to OTU1.

OTU2, retrieves all alarms related to OTU2

OTU3, retrieves all alarms related to OTU3.

OTU1E, retrieves all alarms related to OTU1E.

OTU2E, retrieves all alarms related to OTU2E.

ODU1, retrieves all alarms related to ODU1.

ODUTCM1, retrieves all alarms related to
ODUTCM1.

ODUTCM1E, retrieves all alarms related to
ODUTCM1E.

ODU2, retrieves all alarms related to ODU2.

ODUTCM2, retrieves all alarms related to
ODUTCM2.

ODUTCM2E, retrieves all alarms related to
ODUTCM2E.

ODU3, retrieves all alarms related to ODU3.

ODUTCM3, retrieves all alarms related to
ODUTCM3.

ODU1E, retrieves all alarms related to ODU1E.

ODU2E, retrieves all alarms related to ODU2E.

OPU1, retrieves all alarms related to OPU1.

OPU2, retrieves all alarms related to OPU2.

OPU3, retrieves all alarms related to OPU3.

OPU1E, retrieves all alarms related to OPU1E.

OPU2E, retrieves all alarms related to OPU2E.

SDTRtd, retrieves all alarms related to STD/RTD.

:FETCh[1..n]:DATA:TELecom:ALARm:HISTory?

GFP, retrieves all alarms related to GFP.
ETH, retrieves all alarms related to ETH.
OTU1F, retrieves all alarms related to OTU1F.
OTU2F, retrieves all alarms related to OTU2F.
ODU1F, retrieves all alarms related to ODU1F.
ODU2F, retrieves all alarms related to ODU2F.
ODUTCM1F, retrieves all alarms related to ODUTCM1F.
ODUTCM2F, retrieves all alarms related to ODUTCM1F.
OPU1F, retrieves all alarms related to OPU1F.
OPU2F, retrieves all alarms related to OPU2F.
ALL, retrieves all alarms related to all the tests.

Response Syntax <History>

Response(s) History:
The response data syntax for <History> is defined as a <STRING RESPONSE DATA> element.
Returns the alarm history status of the selected test or all the tests.
PRESENT, indicates that at least one alarm/error has occurred.
ABSENT, indicates that no alarm/error occurred.
INACTIVE, indicates that the test did not run yet.

:FETCh[1..n]:DATA:TELecom:ALARm:HISTory?**Example(s)**

* FETC:DATA:TEL:ALAR:HIST? LOP Returns the history status of all LOP alarms.

Note

SECTion|LINE|PATtern|PORT|RS|MS|HOP|SDT|ALL|ODU3,ODUTCM3|OTU3,OPU3-are supported by 8140 Only

8120,8120NG,8120NGE-|OTU1,ODU1,OPU1,ODUTCM1|SECTion|LINE|PATtern|PORT|E1|E2|E3|E4|RS|MS|LOP|HOP|DS1|DS3|SDT|ALL

8105,8115-SECTion|LINE|PATtern|PORT|E1|E2|E3|E4|RS|MS|LOP|HOP|DS1|DS3|SDT|ALL

8130,8130NG,8130NGE-SECTion|LINE|HOP|PATtern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|OTU2||ODU1|ODUTCM1|ODU2|ODUTCM2|OPU1|OPU2||SDT|ALL

See Also

* FETCh[1..n]:DATA:TELecom:ALARm:CURRent?

* FETCh[1..n]:DATA:TELecom:ALARm:SEConds?

:FETCh[1..n]:DATA:TELEcom:ALARm:SEConds?

Description

This query returns the number of seconds within which all alarms related to the tests such as Port, Pattern, SDT/RTD, OTN, SONET, SDH, DS_n, and PDH occurred.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:ALARm:SEConds?  
<wsp>SECTion|LINE|HOP|PATtern|PORT|E1|  
E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|OTU2|  
OTU3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUT  
CM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|OD  
UTCM3|ODU1E|ODU2E|OPU1|OPU2|OPU3|OP  
U1E|OPU2E|SDT|GFP|ETH|OTU1F|OTU2F|OD  
U1F|ODU2F|ODUTCM1F|ODUTCM2F|OPU1F|O  
PU2F|ALL
```

:FETCh[1..n]:DATA:TELEcom:ALARm:SECOnds?**Parameter(s)**

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

SECTIon|LINE|HOP|PATTern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|OTU2|OTU3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|OPU1|OPU2|OPU3|OPU1E|OPU2E|SDT|GFP|ETH|OTU1F|OTU2F|ODU1F|ODU2F|ODUTCM1F|ODUTCM2F|OPU1F|OPU2F|ALL.

Selects all alarms related to the selected test or all the tests.

SECTIon, retrieves all alarms related to Section.

LINE, retrieves all alarms related to Line.

HOP, retrieves all alarms related to HOP.

PATTern, retrieves all alarms related to Pattern.

PORT, retrieves all alarms related to Port.

E1, retrieves all alarms related to E1.

E2, retrieves all alarms related to E2.

E3, retrieves all alarms related to E3.

E4, retrieves all alarms related to E4.

RS, retrieves all alarms related to RS.

MS, retrieves all alarms related to MS.

LOP, retrieves all alarms related to LOP.

DS1, retrieves all alarms related to DS1.

:FETCH[1..n]:DATA:TELEcom:ALARm:SECOnds?

DS3, retrieves all alarms related to DS3.

OTU1, retrieves all alarms related to OTU1.

OTU2, retrieves all alarms related to OTU2.

OTU3, retrieves all alarms related to OTU3.

OTU1E, retrieves all alarms related to OTU1E.

OTU2E, retrieves all alarms related to OTU2E.

ODU1, retrieves all alarms related to ODU1.

ODUTCM1, retrieves all alarms related to ODUTCM1.

ODUTCM1E, retrieves all alarms related to ODUTCM1E.

ODU2, retrieves all alarms related to ODU2.

ODUTCM2, retrieves all alarms related to ODUTCM2.

ODUTCM2E, retrieves all alarms related to ODUTCM2E.

ODU3, retrieves all alarms related to ODU3.

ODUTCM3, retrieves all alarms related to ODUTCM3.

ODU1E, retrieves all alarms related to ODU1E.

ODU2E, retrieves all alarms related to ODU2E.

OPU1, retrieves all alarms related to OPU1.

OPU2, retrieves all alarms related to OPU2.

OPU3, retrieves all alarms related to OPU3.

OPU1E, retrieves all alarms related to OPU1E.

OPU2E, retrieves all alarms related to OPU2E.

SDTRtd, retrieves all alarms related to STD/RTD.

:FETCh[1..n]:DATA:TELEcom:ALARm:SECOnds?

GFP, retrieves all alarms related to GFP.

ETH, retrieves all alarms related to ETH.

OTU1F, retrieves all alarms related to OTU1F.

OTU2F, retrieves all alarms related to OTU2F.

ODU1F, retrieves all alarms related to ODU1F.

ODU2F, retrieves all alarms related to ODU2F.

ODUTCM1F, retrieves all alarms related to ODUTCM1F.

ODUTCM2F, retrieves all alarms related to ODUTCM1F.

OPU1F, retrieves all alarms related to OPU1F.

OPU2F, retrieves all alarms related to OPU2F.

ALL, retrieves all alarms related to all the tests.

Response Syntax <Seconds>

Response(s) Seconds:

The response data syntax for <Seconds> is defined as a <STRING RESPONSE DATA> element.

Returns the number of seconds within which the selected test alarms or all the test alarms occurred.

:FETCh[1..n]:DATA:TELEcom:ALARm:SECOnds?

Example(s) * FETC:DATA:TEL:ALAR:SEC? LOP Returns the number of seconds within which all LOP alarms occurred.

Note

Only **FTB/IQS-8140** supports
SECTion|LINE|PATtern|PORT|RS|MS|HOP|SDT|ALL|ODU3|ODUTCM3|OTU3|OPU3

Only **FTB/IQS-8120,8120NG,8120NGE** supports
OTU1|ODU1|OPU1|ODUTCM1|SECTion|LINE|PATtern|PORT|E1|E2|E3|E4|RS|MS|LOP|HOP|DS1|DS3|SDT|ALL

Only **FTB/IQS-8105/8115** supports
SECTion|LINE|PATtern|PORT|E1|E2|E3|E4|RS|MS|LOP|HOP|DS1|DS3|SDT|ALL

Only **FTB/IQS-8130/8130NG/8130NGE** supports
SECTion|LINE|HOP|PATtern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|OTU2|ODU1|ODUTCM1|ODU2|ODUTCM2|OPU1|OPU2|SDT|ALL

See Also

- * FETCh[1..n]:DATA:TELEcom:ALARm:HISTory?
- * FETCh[1..n]:DATA:TELEcom:ALARm:CURREnt?

:FETCh[1..n]:DATA:TELEcom:ALARm:CURRent?**Description**

This query returns the current status of all alarms related to the tests such as Port, Pattern, SDT/RTD, OTN, SONET, SDH, DS_n, and PDH. It also returns the combined status of all the tests.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:ALARm:SEConds?
<wsp>SECTion|LINE|HOP|PATtern|PORT|E1|
E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|OTU2|
OTU3|OTU1E|OTU2E|ODU1|ODUTCM1|
ODUTCM1E|ODU2|ODUTCM2|ODUTCM2E|
ODU3|ODUTCM3|ODU1E|ODU2E|OPU1|
OPU2|OPU3|OPU1E|OPU2E|SDT|GFP|ETH|
OTU1F|OTU2F|ODU1F|ODU2F|ODUTCM1F|
ODUTCM2F|OPU1F|OPU2F|ALL
```

:FETCh[1..n]:DATA:TELEcom:ALARm:CURRent?

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

SECTIon|LINE|HOP|PATTern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|OTU2|OTU3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|OPU1|OPU2|OPU3|OPU1E|OPU2E|SDT|GFP|ETH|OTU1F|OTU2F|ODU1F|ODU2F|ODUTCM1F|ODUTCM2F|OPU1F|OPU2F|ALL

Selects all alarms related to the selected test or all the tests.

SECTIon, retrieves all alarms related to Section.

LINE, retrieves all alarms related to Line.

HOP, retrieves all alarms related to HOP.

PATTern, retrieves all alarms related to Pattern.

PORT, retrieves all alarms related to Port.

E1, retrieves all alarms related to E1.

E2, retrieves all alarms related to E2.

E3, retrieves all alarms related to E3.

E4, retrieves all alarms related to E4.

RS, retrieves all alarms related to RS.

MS, retrieves all alarms related to MS.

LOP, retrieves all alarms related to LOP.

DS1, retrieves all alarms related to DS1.

:FETCh[1..n]:DATA:TELeom:ALARm:CURRent?

DS3, retrieves all alarms related to DS3.

OTU1, retrieves all alarms related to OTU1.

OTU2, retrieves all alarms related to OTU2.

OTU3, retrieves all alarms related to OTU3.

OTU1E, retrieves all alarms related to OTU1E.

OTU2E, retrieves all alarms related to OTU2E.

ODU1, retrieves all alarms related to ODU1.

ODUTCM1, retrieves all alarms related to ODUTCM1.

ODUTCM1E, retrieves all alarms related to ODUTCM1E.

ODU2, retrieves all alarms related to ODU2.

ODUTCM2, retrieves all alarms related to ODUTCM2.

ODUTCM2E, retrieves all alarms related to ODUTCM2E.

ODU3, retrieves all alarms related to ODU3.

ODUTCM3, retrieves all alarms related to ODUTCM3.

ODU1E, retrieves all alarms related to ODU1E.

ODU2E, retrieves all alarms related to ODU2E.

OPU1, retrieves all alarms related to OPU1.

OPU2, retrieves all alarms related to OPU2.

OPU3, retrieves all alarms related to OPU3.

OPU1E, retrieves all alarms related to OPU1E.

OPU2E, retrieves all alarms related to OPU2E.

SDTRtd, retrieves all alarms related to STD/RTD.

:FETCh[1..n]:DATA:TELEcom:ALARm:CURRent?

GFP, retrieves all alarms related to GFP.

ETH, retrieves all alarms related to ETH.

OTU1F, retrieves all alarms related to OTU1F.

OTU2F, retrieves all alarms related to OTU2F.

ODU1F, retrieves all alarms related to ODU1F.

ODU2F, retrieves all alarms related to ODU2F.

ODUTCM1F, retrieves all alarms related to ODUTCM1F.

ODUTCM2F, retrieves all alarms related to ODUTCM1F.

OPU1F, retrieves all alarms related to OPU1F.

OPU2F, retrieves all alarms related to OPU2F.

ALL, retrieves all alarms related to all the tests.

Response Syntax <Current>

:FETCh[1..n]:DATA:TELecom:ALARm:CURRent?

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the current alarm status of the selected test or all the tests.</p>
Example(s)	<p>* FETC:DATA:TEL:ALAR:CURR? LOP Returns the current status of all LOP alarms.</p>
See Also	<p>* FETCh[1..n]:DATA:TELecom:ALARm:HISTory?</p> <p>* FETCh[1..n]:DATA:TELecom:ALARm:SECConds?</p>

:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?

Description

This query returns the history status of all errors related to the tests such as Port, Pattern, OTN, SONET, SDH, DS_n, and PDH. It also returns the combined status of all the tests.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?

<wsp>SECTion|LINE|HOP|PATtern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|FEC1|OTU2|FEC2|OTU3|FEC3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|FEC1E|FEC2E|GFP|ETH|ALL|OTU1F|OTU2F|ODU1F|ODU2F|ODUTCM1F|ODUTCM2F|FEC1F|FEC2F

:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?**Parameter(s)**

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

SECTIon|LINE|HOP|PATTern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|FEC1|OTU2|FEC2|OTU3|FEC3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|FEC1E|FEC2E|GFP|ETH|ALL|OTU1F|OTU2F|ODU1F|ODU2F|ODUTCM1F|ODUTCM2F|FEC1F|FEC2F.

Selects all errors related to the selected test or all the tests.

SECTIon, retrieves all errors related to Section.

LINE, retrieves all errors related to Line.

HOP, retrieves all errors related to HOP.

PATTern, retrieves all errors related to Pattern.

PORT, retrieves all errors related to Port.

E1, retrieves all errors related to E1.

E2, retrieves all errors related to E2.

E3, retrieves all errors related to E3.

E4, retrieves all errors related to E4.

RS, retrieves all errors related to RS.

MS, retrieves all errors related to MS.

LOP, retrieves all errors related to LOP.

DS1, retrieves all errors related to DS1.

:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?

DS3, retrieves all errors related to DS3.

OUT1, retrieves all errors related to OTU1.

FEC1, retrieves all errors related to FEC1.

OTU2, retrieves all errors related to OTU2

FEC2, retrieves all errors related to FEC2.

OTU3, retrieves all errors related to OTU3.

FEC3, retrieves all errors related to FEC3.

OTU1E, retrieves all errors related to OTU1E.

OTU2E, retrieves all errors related to OTU2E.

ODU1, retrieves all errors related to ODU1.

ODUTCM1, retrieves all errors related to
ODUTCM1.

ODUTCM1E, retrieves all errors related to
ODUTCM1E.

ODU2, retrieves all errors related to ODU2.

ODUTCM2, retrieves all errors related to
ODUTCM2.

ODUTCM2E, retrieves all errors related to
ODUTCM2E.

ODU3, retrieves all errors related to ODU3.

ODUTCM3, retrieves all errors related to
ODUTCM3.

ODU1E, retrieves all errors related to ODU1E.

ODU2E, retrieves all errors related to ODU2E.

FEC1E, retrieves all errors related to FEC1E.

FEC2E, retrieves all errors related to FEC2E.

GFP, retrieves all errors related to GFP.

:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?

ETH, retrieves all errors related to ETH.

OTU1F, retrieves all errors related to OTU1F.

OTU2F, retrieves all errors related to OTU2F.

ODU1F, retrieves all errors related to ODU1F.

ODU2F, retrieves all errors related to ODU2F.

ODUTCM1F, retrieves all errors related to ODUTCM1F.

ODUTCM2F, retrieves all errors related to ODUTCM2F.

FEC1F, retrieves all errors related to FEC1F.

FEC2F, retrieves all errors related to FEC2F.

ALL, retrieves all alarms related to all the tests.

Response Syntax <History>

Response(s)

History:

The response data syntax for <History> is defined as a <STRING RESPONSE DATA> element.

Returns the errored history status of the selected test or all the tests.

:FETCh[1..n]:DATA:TELEcom:ERRor:HISTory?

Example(s) * FETC:DATA:TEL:ERR:HIST? LOP Returns the history status of all LOP errors.

Note SECTION|LINE|PATtern|PORT|RS|MS|HOP|SDT|ALL|ODU3,ODUTCM3|OTU3,OPU3|FEC3-are supported by 8140 Only
8120,8120NG,8120NGE-FEC1|OTU1,ODU1,OPU1,ODUTCM1|SECTION|LINE|PATtern|PORT|E1|E2|E3|E4|RS|MS|LOP|HOP|DS1|DS3|SDT|ALL
8105,8115-SECTION|LINE|PATtern|PORT|E1|E2|E3|E4|RS|MS|LOP|HOP|DS1|DS3|SDT|ALL
8130,8130NG,8130NGE-SECTION|LINE|HOP|PATtern|PORT|E1|E2|E3|E4|RS|FEC1|FEC2|MS|LOP|DS1|DS3|OTU1|OTU2||ODU1|ODUTCM1|ODU2|ODUTCM2||SDT|ALL

See Also * FETCh[1..n]:DATA:TELEcom:ERRor:CURRent?
* FETCh[1..n]:DATA:TELEcom:ERRor:SECOnds?
* FETCh[1..n]:DATA:TELEcom:ERRor:COUnT?
* FETCh[1..n]:DATA:TELEcom:ERRor:RATE?

:FETCh[1..n]:DATA:TELEcom:ERRor:SEConds?**Description**

This query returns the number of seconds within which all errors related to the tests such as Port, Pattern, OTN, SONET, SDH, DS_n, and PDH occurred.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:ERRor:SEConds?
<wsp>SECTion|LINE|HOP|PATtern|PORT|E1|
E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|FEC1|
OTU2|FEC2|OTU3|FEC3|OTU1E|OTU2E|
DU1|ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|
ODUTCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|
FEC1E|FEC2E|GFP|ETH|ALL|OTU1F|OTU2F|
ODU1F|ODU2F|ODUTCM1F|ODUTCM2|
FFEC1F|FEC2F
```

:FETCh[1..n]:DATA:TELecom:ERRor:SECOnds?**Parameter(s)**

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

SECTioN|LINE|HOP|PATTern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|FEC1|OTU2|FEC2|OTU3|FEC3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|FEC1E|FEC2E|GFP|ETH|ALL|OTU1F|OTU2F|ODU1F|ODU2F|ODUTCM1F|ODUTCM2F|FEC1F|FEC2F.

Selects all errors related to the selected test or all the tests.

SECTioN, retrieves all errors related to Section.

LINE, retrieves all errors related to Line.

HOP, retrieves all errors related to HOP.

PATTern, retrieves all errors related to Pattern.

PORT, retrieves all errors related to Port.

E1, retrieves all errors related to E1.

E2, retrieves all errors related to E2.

E3, retrieves all errors related to E3.

E4, retrieves all errors related to E4.

RS, retrieves all errors related to RS.

MS, retrieves all errors related to MS.

LOP, retrieves all errors related to LOP.

DS1, retrieves all errors related to DS1.

:FETCh[1..n]:DATA:TELecom:ERRor:SEConds?

DS3, retrieves all errors related to DS3.

OUT1, retrieves all errors related to OTU1.

FEC1, retrieves all errors related to FEC1.

OTU2, retrieves all errors related to OTU2

FEC2, retrieves all errors related to FEC2.

OTU3, retrieves all errors related to OTU3.

FEC3, retrieves all errors related to FEC3.

OTU1E, retrieves all errors related to OTU1E.

OTU2E, retrieves all errors related to OTU2E.

ODU1, retrieves all errors related to ODU1.

ODUTCM1, retrieves all errors related to
ODUTCM1.

ODUTCM1E, retrieves all errors related to
ODUTCM1E.

ODU2, retrieves all errors related to ODU2.

ODUTCM2, retrieves all errors related to
ODUTCM2.

ODUTCM2E, retrieves all errors related to
ODUTCM2E.

ODU3, retrieves all errors related to ODU3.

ODUTCM3, retrieves all errors related to
ODUTCM3.

ODU1E, retrieves all errors related to ODU1E.

ODU2E, retrieves all errors related to ODU2E.

FEC1E, retrieves all errors related to FEC1E.

FEC2E, retrieves all errors related to FEC2E.

GFP, retrieves all errors related to GFP.

:FETCh[1..n]:DATA:TELecom:ERRor:SEConds?

ETH, retrieves all errors related to ETH.

OTU1F, retrieves all errors related to OTU1F.

OTU2F, retrieves all errors related to OTU2F.

ODU1F, retrieves all errors related to ODU1F.

ODU2F, retrieves all errors related to ODU2F.

ODUTCM1F, retrieves all errors related to
ODUTCM1F.

ODUTCM2F, retrieves all errors related to
ODUTCM2F.

FEC1F, retrieves all errors related to FEC1F.

FEC2F, retrieves all errors related to FEC12F.

ALL, retrieves all alarms related to all the tests.

Response Syntax <Seconds>

:FETCh[1..n]:DATA:TELecom:ERRor:SECOnds?

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <STRING RESPONSE DATA> element. Returns the number of seconds within which the selected test errors or all the test errors occurred.
Example(s)	* FETC:DATA:TEL:ERR:SEC? LOP Returns the number of seconds within which all LOP errors occurred.
See Also	* FETCh[1..n]:DATA:TELecom:ERRor:HISTory? * FETCh[1..n]:DATA:TELecom:ERRor:CURRent?

:FETCh[1..n]:DATA:TELEcom:ERRor:CURRent?

Description

This query returns the current status of all errors related to the tests such as Port, Pattern, OTN, SONET, SDH, DS_n, and PDH. It also returns the combined status of all the tests.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:ERRor:CURRent?  
<wsp>SECTion|LINE|HOP|PATtern|PORT|E1|  
E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|FEC1|O  
TU2|FEC2|OTU3|FEC3|OTU1E|OTU2E|ODU1|  
ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODU  
TCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|FEC1  
E|FEC2E|GFP|ETH|ALL|OTU1F|OTU2F|0DU1F  
|ODU2F|ODUTCM1F|ODUTCM2F|FEC1F|FEC2F
```

:FETCh[1..n]:DATA:TELEcom:ERRor:CURRent?**Parameter(s)**

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

SECTion|LINE|HOP|PATTern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|FEC1|OTU2|FEC2|OTU3|FEC3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|FEC1E|FEC2E|GFP|ETH|ALL|OTU1F|OTU2F|ODU1F|ODU2F|ODUTCM1F|ODUTCM2F|FEC1F|FEC1F.

Selects all errors related to the selected test or all the tests.

SECTion, retrieves all errors related to Section.

LINE, retrieves all errors related to Line.

HOP, retrieves all errors related to HOP.

PATTern, retrieves all errors related to Pattern.

PORT, retrieves all errors related to Port.

E1, retrieves all errors related to E1.

E2, retrieves all errors related to E2.

E3, retrieves all errors related to E3.

E4, retrieves all errors related to E4.

RS, retrieves all errors related to RS.

MS, retrieves all errors related to MS.

LOP, retrieves all errors related to LOP.

DS1, retrieves all errors related to DS1.

:FETCh[1..n]:DATA:TELeCom:ERRor:CURRent?

DS3, retrieves all errors related to DS3.

OUT1, retrieves all errors related to OTU1.

FEC1, retrieves all errors related to FEC1.

OTU2, retrieves all errors related to OTU2

FEC2, retrieves all errors related to FEC2.

OTU3, retrieves all errors related to OTU3.

FEC3, retrieves all errors related to FEC3.

OTU1E, retrieves all errors related to OTU1E.

OTU2E, retrieves all errors related to OTU2E.

ODU1, retrieves all errors related to ODU1.

ODUTCM1, retrieves all errors related to
ODUTCM1.

:FETCh[1..n]:DATA:TELecom:ERRor:CURRent?

ODUTCM1E, retrieves all errors related to ODUTCM1E.

ODU2, retrieves all errors related to ODU2.

ODUTCM2, retrieves all errors related to ODUTCM2.

ODUTCM2E, retrieves all errors related to ODUTCM2E.

ODU3, retrieves all errors related to ODU3.

ODUTCM3, retrieves all errors related to ODUTCM3.

ODU1E, retrieves all errors related to ODU1E.

ODU2E, retrieves all errors related to ODU2E.

FEC1E, retrieves all errors related to FEC1E.

FEC2E, retrieves all errors related to FEC2E.

GFP, retrieves all errors related to GFP.

ETH, retrieves all errors related to ETH.

OTU1F, retrieves all errors related to OTU1F.

OTU2F, retrieves all errors related to OTU2F.

ODU1F, retrieves all errors related to ODU1F.

ODU2F, retrieves all errors related to ODU2F.

ODUTCM1F, retrieves all errors related to ODUTCM1F.

ODUTCM2F, retrieves all errors related to ODUTCM2F.

FEC1F, retrieves all errors related to FEC1F.

FEC2F, retrieves all errors related to FEC12F.

:FETCh[1..n]:DATA:TELecom:ERRor:CURRent?

Response Syntax <Current>

Response(s)

Current:

The response data syntax for <Current> is defined as a <STRING RESPONSE DATA> element.

Returns the current errored status of the selected test or all the tests.

Example(s)

* FETC:DATA:TEL:ERR:CURR? LOP Returns the current status of all LOP errors.

See Also

- * FETCh[1..n]:DATA:TELecom:ERRor:HISTory?
- * FETCh[1..n]:DATA:TELecom:ERRor:SECOnds?
- * FETCh[1..n]:DATA:TELecom:ERRor:COUNt?
- * FETCh[1..n]:DATA:TELecom:ERRor:RATE?

:FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?**Description**

This query returns the count of all errors related to the tests such as Port, Pattern, OTN, SONET, SDH, DS_n, and PDH.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?  
<wsp>SECTion|LINE|HOP|PATtern|PORT|E1|  
E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|FEC1|O  
TU2|FEC2|OTU3|FEC3|OTU1E|OTU2E|ODU1|  
ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODU  
TCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|FEC1  
E|FEC2E|GFP|ETH|ALL|OTU1F|OTU2F|0DU1F  
|ODU2F|ODUTCM1F|ODUTCM2F|FEC1F|FEC2F
```

:FETCh[1..n]:DATA:TELecom:ERRor:COUnT?

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

SECTIon|LINE|HOP|PATTern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|FEC1|OTU2|FEC2|OTU3|FEC3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|FEC1E|FEC2E|GFP|ETH|ALL|OTU1F|OTU2F|ODU1F|ODU2F|ODUTCM1F|ODUTCM2F|FEC1F|FEC2F.

Selects all errors related to the selected test or all the tests.

SECTIon, retrieves all errors related to Section.

LINE, retrieves all errors related to Line.

HOP, retrieves all errors related to HOP.

PATTern, retrieves all errors related to Pattern.

PORT, retrieves all errors related to Port.

E1, retrieves all errors related to E1.

E2, retrieves all errors related to E2.

E3, retrieves all errors related to E3.

E4, retrieves all errors related to E4.

RS, retrieves all errors related to RS.

MS, retrieves all errors related to MS.

LOP, retrieves all errors related to LOP.

DS1, retrieves all errors related to DS1.

:FETCh[1..n]:DATA:TELeCom:ERRor:COUnT?

DS3, retrieves all errors related to DS3.

OUT1, retrieves all errors related to OTU1.

FEC1, retrieves all errors related to FEC1.

OTU2, retrieves all errors related to OTU2

FEC2, retrieves all errors related to FEC2.

OTU3, retrieves all errors related to OTU3.

FEC3, retrieves all errors related to FEC3.

OTU1E, retrieves all errors related to OTU1E.

OTU2E, retrieves all errors related to OTU2E.

ODU1, retrieves all errors related to ODU1.

ODUTCM1, retrieves all errors related to
ODUTCM1.

ODUTCM1E, retrieves all errors related to
ODUTCM1E.

ODU2, retrieves all errors related to ODU2.

ODUTCM2, retrieves all errors related to
ODUTCM2.

ODUTCM2E, retrieves all errors related to
ODUTCM2E.

ODU3, retrieves all errors related to ODU3.

ODUTCM3, retrieves all errors related to
ODUTCM3.

ODU1E, retrieves all errors related to ODU1E.

ODU2E, retrieves all errors related to ODU2E.

FEC1E, retrieves all errors related to FEC1E.

FEC2E, retrieves all errors related to FEC2E.

GFP, retrieves all errors related to GFP.

:FETCh[1..n]:DATA:TELecom:ERRor:COUNT?

ETH, retrieves all errors related to ETH.

OTU1F, retrieves all errors related to OTU1F.

OTU2F, retrieves all errors related to OTU2F.

ODU1F, retrieves all errors related to ODU1F.

ODU2F, retrieves all errors related to ODU2F.

ODUTCM1F, retrieves all errors related to
ODUTCM1F.

ODUTCM2F, retrieves all errors related to
ODUTCM2F.

FEC1F, retrieves all errors related to FEC1F.

FEC2F, retrieves all errors related to FEC12F.

Response Syntax <Count>

:FETCh[1..n]:DATA:TELEcom:ERRor:COUNT?

Response(s)	Count: The response data syntax for <Count> is defined as a <STRING RESPONSE DATA> element. Returns the error count for the selected test errors and all the test errors.
Example(s)	* FETC:DATA:TEL:ERR:COUN? LOP Returns the count of all LOP errors.
See Also	* 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?

Description

This query returns the current rate of all errors related to the tests such as Port, Pattern, OTN, SONET, SDH, DS_n, and PDH.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:ERRor:RATE?  
<wsp>SECTion|LINE|HOP|PATtern|PORT|E1|  
E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|FEC1|O  
TU2|FEC2|OTU3|FEC3|OTU1E|OTU2E|ODU1|  
ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODU  
TCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|FEC1  
E|FEC2E|GFP|ETH|ALL|OTU1F|OTU2F|0DU1F  
|ODU2F|ODUTCM1F|ODUTCM2F|FEC1F|FEC2F
```

:FETCh[1..n]:DATA:TELEcom:ERRor:RATE?**Parameter(s)**

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

SECTIon|LINE|HOP|PATTern|PORT|E1|E2|E3|E4|RS|MS|LOP|DS1|DS3|OTU1|FEC1|OTU2|FEC2|OTU3|FEC3|OTU1E|OTU2E|ODU1|ODUTCM1|ODUTCM1E|ODU2|ODUTCM2|ODUTCM2E|ODU3|ODUTCM3|ODU1E|ODU2E|FEC1E|FEC2E|GFP|ETH|ALL|OTU1F|OTU2F|ODU1F|ODU2F|ODUTCM1F|ODUTCM2F|FEC1F|FEC2F.

Selects all errors related to the selected test or all the tests.

SECTIon, retrieves all errors related to Section.

LINE, retrieves all errors related to Line.

HOP, retrieves all errors related to HOP.

PATTern, retrieves all errors related to Pattern.

PORT, retrieves all errors related to Port.

E1, retrieves all errors related to E1.

E2, retrieves all errors related to E2.

E3, retrieves all errors related to E3.

E4, retrieves all errors related to E4.

RS, retrieves all errors related to RS.

MS, retrieves all errors related to MS.

LOP, retrieves all errors related to LOP.

DS1, retrieves all errors related to DS1.

:FETCh[1..n]:DATA:TELEcom:ERRor:RATE?

DS3, retrieves all errors related to DS3.
OUT1, retrieves all errors related to OTU1.
FEC1, retrieves all errors related to FEC1.
OTU2, retrieves all errors related to OTU2
FEC2, retrieves all errors related to FEC2.
OTU3, retrieves all errors related to OTU3.
FEC3, retrieves all errors related to FEC3.
OTU1E, retrieves all errors related to OTU1E.
OTU2E, retrieves all errors related to OTU2E.
ODU1, retrieves all errors related to ODU1.
ODUTCM1, retrieves all errors related to
ODUTCM1.
ODUTCM1E, retrieves all errors related to
ODUTCM1E.
ODU2, retrieves all errors related to ODU2.
ODUTCM2, retrieves all errors related to
ODUTCM2.
ODUTCM2E, retrieves all errors related to
ODUTCM2E.
ODU3, retrieves all errors related to ODU3.
ODUTCM3, retrieves all errors related to
ODUTCM3.
ODU1E, retrieves all errors related to ODU1E.
ODU2E, retrieves all errors related to ODU2E.
FEC1E, retrieves all errors related to FEC1E.
FEC2E, retrieves all errors related to FEC2E.
GFP, retrieves all errors related to GFP.

:FETCh[1..n]:DATA:TELEcom:ERRor:RATE?

ETH, retrieves all errors related to ETH.

OTU1F, retrieves all errors related to OTU1F.

OTU2F, retrieves all errors related to OTU2F.

ODU1F, retrieves all errors related to ODU1F.

ODU2F, retrieves all errors related to ODU2F.

ODUTCM1F, retrieves all errors related to ODUTCM1F.

ODUTCM2F, retrieves all errors related to ODUTCM2F.

FEC1F, retrieves all errors related to FEC1F.

FEC2F, retrieves all errors related to FEC12F.

Response Syntax <Rate>

Response(s) Rate:
The response data syntax for <Rate> is defined as a <STRING RESPONSE DATA> element.
Returns the current error rate for the selected test errors or all the test errors.

:FETCh[1..n]:DATA:TELEcom:ERRor:RATE?

Example(s) * FETC:DATA:TEL:ERR:RATE? LOP Returns the current error rate of all LOP errors.

Note

Only **FTB/IQS-8140** supports
SECTion|LINE|PATtern|PORT|RS|MS|LOP|
HOP|SDT|ALL|ODU3,ODUTCM3|OTU3,OPU3

Only **FTB/IQS-8120,8120NG,8120NGE** supports
FEC1|OTU1,ODU1,OPU1,ODUTCM1|SECTion|
LINE|PATtern|PORT|E1|E2|E3|E4|RS|MS|LOP
|HOP|DS1|SDT|ALL

Only **FTB/IQS-8105/8115** supports
SECTion|LINE|PATtern|PORT|E1|E2|E3|E4|RS
|MS|LOP|HOP|DS1|SDT|ALL

Only **FTB/IQS-8130/8130NG/8130NGE** supports
SECTion|LINE|HOP|PATtern|PORT|E1|E2|E3|
E4|RS|FEC1|FEC2|MS|LOP|DS1|DS3|OTU1|
OTU2|ODU1|ODUTCM1|ODU2|ODUTCM2|
OPU1|OPU2|SDT|ALL

See Also

- * 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?

Logger Command Reference

:FETCh[1..n]:DATA:TELecom:LOGGer:EVENTs?

Description	This query returns the total number of test events recorded. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELecom:LOGGer:EVENTs?
Parameter(s)	None
Response Syntax	<Event>
Response(s)	Event: The response data syntax for <Event> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the total number of test events recorded.
Example(s)	* FETC:DATA:TEL:LOGG:EVEN? Returns the total number of test events recorded.
See Also	* FETCh[1..n]:DATA:TELecom:LOGGer:LIST?

:FETCh[1..n]:DATA:TELEcom:LOGGer:LIST?

Description This query returns the list of test events.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:LOGGer:LIST?
<wsp>MAXimum|MINimum

Parameter(s) Eventno:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum|MINimum
This parameter is optional. If no token is specified, the entire event list will be returned.

Response Syntax <List>

:FETCh[1..n]:DATA:TELecom:LOGGer:LIST?

Response(s)	List: The response data syntax for <List> is defined as a <STRING RESPONSE DATA> element. Returns the list of test events.
Example(s)	* FETC:DATA:TEL:LOGG:LIST? 2 Returns the list of test events.
See Also	* FETCh[1..n]:DATA:TELecom:LOGGer:EVENts?

GFP Commands

:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: COUNT:TX?

Description	<p>This query returns the count of transmitted frames.</p> <p>At *RST, this value is 0.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: COUNT:TX? <wsp>DFRames MFRames IFRames TFRames</p>
Parameter(s)	<p>FRAMES:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames IFRames TFRames.</p> <p>Selects the the the count of transmitted frames.</p> <p>DFRames, selects Data Frames as transmitted frame.</p> <p>MFRames, selects Management Frames as transmitted frame.</p> <p>IFRames, selects Idle Frames as transmitted frame.</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:
COUNT:TX?****Response Syntax** <Frames>**Response(s)**

Frames:

The response data syntax for <Frames> is defined as a <NR2 NUMERIC RESPONSE DATA> element.

Returns the frame counts.

Example(s)FETC:DATA:TEL:GFP:OVER:COUN:TX? DFRames

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:
RATE:TX?**

Description	<p>This query returns the Rate of transmitted frames.</p> <p>At *RST, this value is 0.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: RATE:TX? <wsp>DFRames MFRames IFRames TFRames</p>
Parameter(s)	<p>FRAMES:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames IFRames TFRames.</p> <p>Selects the Rate of transmitted frames.</p> <p>DFRames, selects Data Frames as transmitted frame.</p> <p>MFRames, selects Management Frames as transmitted frame.</p> <p>IFRames, selects Idle Frames as transmitted frame.</p> <p>TFRames, selects TFRames as valid frame count.</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:
RATE:TX?****Response Syntax** <Frames>**Response(s)** Frames:
The response data syntax for <Frames> is defined as a <NR3 NUMERIC RESPONSE DATA> element.
Returns the Rate of transmitted frames.**Example(s)** FETC:DATA:TEL:GFP:OVER:RATE:TX? DFRames

:FETCh[1..n]:DATA:TELEcom:GFP:CStat:COUNT:TX?

Description	<p>This query returns the count of the transmitted Client Data Frames and Client Management Frames.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:GFP:CStat:COUNT:TX? <wsp>DFRmes MFRames</p>
Parameter(s)	<p>Frames:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRmes MFRames.</p> <p>Selects the count of the transmitted frames.</p> <p>DFRmes, selects Data Frames as the count of the transmitted frames.</p> <p>MFRames, selects Management Frames as the count of the transmitted frames.</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:CStat:
COUNT:TX?****Response Syntax** <Frames>**Response(s)**

Frames:

The response data syntax for <Frames> is defined as a <NR2 NUMERIC RESPONSE DATA> element.

Returns the count of the transmitted Client Data Frames and Client Management Frames.

Example(s)FETC:DATA:TEL:GFP:CST:COUN:TX? DFRames

**:FETCh[1..n]:DATA:TELEcom:GFP:CStat:
RATE:TX?**

Description

This query returns the rate of the transmitted Client Data Frames and Client Management Frames.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:GFP:CStat:RATE:
TX? <wsp>DFRmes|MFRames

Parameter(s)

Frames:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
DFRmes|MFRames.

Selects the count of the transmitted frames.

DFRmes, selects Data Frames as the count of the transmitted frames.

MFRames, selects Management Frames as the count of the transmitted frames.

**:FETCh[1..n]:DATA:TELEcom:GFP:CStat:
RATE:TX?****Response Syntax** <Frames>**Response(s)**

Frames:

The response data syntax for <Frames> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the rate of the transmitted Client Data Frames and Client Management Frames.

Example(s)

FETC:DATA:TEL:GFP:CST:RATE:TX? DFRames

Returns the rate of the transmitted Client Data Frames and Client Management Frames.

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:
BANDwidth:TX?**

Description	<p>This query returns the percentage of transmitted transport layer Bandwidth in the last second, excluding the idle bytes.</p> <p>At *RST, this value is 0.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: BANDwidth:TX?</code>
Parameter(s)	None
Response Syntax	<code><TX></code>
Response	<p>TX:</p> <p>The response data syntax for <code><TX></code> is defined as a <code><NR2 NUMERIC RESPONSE DATA></code> element.</p> <p>Returns the percentage of transmitted transport layer Bandwidth in the last second, excluding the idle bytes.</p>
Example(s)	<code>FETC:DATA:TEL:GFP:OVER:BAND:TX?</code>

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:
EFFiciency:TX?**

Description	<p>This query returns the transmitted transport layer Mapping Efficiency (Client Payload Bytes divided by Client Data Bytes multiplied by 100) in the last second.</p> <p>At *RST, this value is 0.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: EFFiciency:TX?
Parameter(s)	None
Response Syntax	<TX>
Response	<p>TX:</p> <p>The response data syntax for <TX> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the transmitted transport layer Mapping Efficiency (Client Payload Bytes divided by Client Data Bytes multiplied by 100) in the last second.</p>
Example(s)	FETC:DATA:TEL:GFP:OVER:EFF:TX?

:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: COUNT:RX?

Description This query returns the count of received frames.

At *RST, this value is 0.

Syntax :FETCh[1..n]:DATA:TELEcom:GFP:OVERview:
COUNT:RX? <wsp>DFRames|MFRames|
IFRames|TFRames|RPLiframes|INVframes
|DISframes

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:
COUNT:RX?**

Parameter(s)	<p>FRAMES:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames IFRames TFRames RPLiframes INVFrames DISFrames.</p> <p>Selects the frame counts.</p> <p>DFRames, selects DFRmes as valid frame count.</p> <p>MFRames, selects MFRames as valid frame count.</p> <p>IFRames, selects IFRames as valid frame count.</p> <p>TFRames, selects TFRames as valid frame count.</p> <p>RPLiframes, selects RPLiframes as valid frame count.</p> <p>INVFrames, selects INVFrames as valid frame count.</p> <p>DISFrames, selects DISFrames as valid frame count.</p>
---------------------	--

:FETCh[1..n]:DATA:TELecom:GFP:OVERview: COUNT:RX?

Response Syntax <Frames>

Response(s) Frames:
The response data syntax for <Frames> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of received frames.

Example(s) FETC:DATA:TEL:GFP:OVER:COUN:RX? DFRames

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:
RATE:RX?**

Description	<p>This query returns the Rate of received frames.</p> <p>At *RST, this value is 0.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: RATE:RX? <wsp>DFRames MFRames IFRames TFRames</pre>
Parameter(s)	<p>FRAMES:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames IFRames TFRames.</p> <p>Selects the frame counts.</p> <p>DFRames, selects DFRames as valid frame count.</p> <p>MFRames, selects MFRames as valid frame count.</p> <p>IFRames, selects IFRames as valid frame count.</p> <p>TFRames, selects TFRames as valid frame count.</p> <hr/>

**:FETCh[1..n]:DATA:TELecom:GFP:OVERview:
RATE:RX?**

Response Syntax <Frames>

Response(s) Frames:
The response data syntax for <Frames> is defined as a <NR3 NUMERIC RESPONSE DATA> element.
Returns the Rate of received frames.

Example(s) FETC:DATA:TEL:GFP:OVER:RATE:RX? DFRames

**:FETCh[1..n]:DATA:TELEcom:GFP:CStat:
COUNT:RX?**

Description	<p>This query returns the count of the received Client Data Frames and Client Management Frames.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:GFP:CStat:COUNT:RX? <wsp>DFRmes MFRames RPTiframes</pre>
Parameter(s)	<p>Frames:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRmes MFRames RPTiframes.</p> <p>Selects the count of the received frames.</p> <p>DFRmes, selects Data Frames as the count of the received frames.</p> <p>MFRames, selects Management Frames as the count of the received frames.</p> <p>RPTiframes, selects Reserved PTI Frames as the count of the received frames.</p>

:FETCh[1..n]:DATA:TELEcom:GFP:CStat: COUNT:RX?

Response Syntax <Frames>

Response(s) Frames:
The response data syntax for <Frames> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of the received Client Data Frames and Client Management Frames.

Example(s) FETC:DATA:TEL:GFP:CST:COUN:RX? DFRames
Returns the count of the received Client Data Frames and Client Management Frames.

**:FETCh[1..n]:DATA:TELEcom:GFP:CStat:
RATE:RX?**

Description	<p>This query returns the rate of the received Client Data Frames and Client Management Frames.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:GFP:CStat:RATE:RX? <wsp>DFRmes MFRames RPTiframes</pre>
Parameter(s)	<p>Frames:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRmes MFRames RPTiframes.</p> <p>Selects the count of the received frames.</p> <p>DFRmes, selects Data Frames as the count of the received frames.</p> <p>MFRames, selects Management Frames as the count of the received frames.</p> <p>RPTiframes, selects Reserved PTI Frames as the count of the received frames.</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:CStat:
RATE:RX?**

Response Syntax <Frames>

Response(s) Frames:
The response data syntax for <Frames> is defined as a <NR3 NUMERIC RESPONSE DATA> element.
Returns the rate of the received Client Data Frames and Client Management Frames.

Example(s) FETC:DATA:TEL:GFP:CST:RATE:RX? DFRames
Returns the rate of the received Client Data Frames and Client Management Frames.

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:
BANDwidth:RX?**

Description	<p>This query returns the percentage of Received transport layer Bandwidth in the last second, excluding the idle bytes.</p> <p>At *RST, this value is 0.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: BANDwidth:RX?
Parameter(s)	None
Response Syntax	TX
Response(s)	<p>TX:</p> <p>The response data syntax for <TX> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the percentage of Received transport layer Bandwidth in the last second, excluding the idle bytes.</p>
Example(s)	FETC:DATA:TEL:GFP:OVER:BAND:RX?

**:FETCh[1..n]:DATA:TELEcom:GFP:OVERview:
EFFiciency:RX?**

Description	<p>This query returns the percentage of received transport layer Mapping efficiency in the last second, excluding the idle bytes.</p> <p>At *RST, this value is 0.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:GFP:OVERview: EFFiciency:RX?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><TX></p>
Response(s)	<p>TX:</p> <p>The response data syntax for <TX> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the percentage of Received transport layer Mapping efficiency in the last second, excluding the idle bytes.</p>
Example(s)	<p>FETC:DATA:TEL:GFP:OVER:EFF:RX?</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
HISTory?**

Description	<p>This query returns the history status of alarm analysis.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: HISTory? <wsp>FRAMe CHANnel</p>
Parameter(s)	<p>History:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FRAMe CHANnel</p> <p>Selects the history of alarms.</p> <p>FRAMe, selects the history of FRAMe alarms.</p> <p>CHANnel, selects the history of CHANnel alarms.</p>
Response Syntax	<p><History></p>

:FETCh[1..n]:DATA:TELecom:GFP:ALARm: HISTory?

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <Character response data> element.</p> <p>Returns the history status of alarm analysis.</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>
Example(s)	<p>FETC:DATA:TEL:GFP:ALAR:HIST? FRAME</p>
See Also	<p>FETC[1..n]:DATA:TELecom:GFP:ALARm: CURRent?</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
CURRent?**

Description	<p>This query returns the current status of alarm analysis.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: CURRent? <wsp>FRAME CHANnel</pre>
Parameter(s)	<p>Current:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FRAME CHANnel.</p> <p>Selects the current status of alarm analysis.</p> <p>FRAME, selects the current status of FRAME alarm analysis.</p> <p>CHANnel, selects the current status of CHANnel alarm analysis.</p>
Response Syntax	<pre><Current></pre>

**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <Character response data> element.</p> <p>Returns the current status of alarms.</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>
Example(s)	<p>FETC:DATA:TEL:GFP:ALAR:CURR? FRAMe</p>
See Also	<p>FETC[1..n]:DATA:TELEcom:GFP:ALARm: HISTory?</p>

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

Description	<p>This command sets the configuration of Extension Header Identifier (EHI). Allows the selection of GFP Extension Header.</p> <p>At *RST, this value is Null.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:GFP: CONFiguration:EXI<wsp>NULL LINear</pre>
Parameter(s)	<p>EXI:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: NULL LINear.</p> <p>Selects the configuration of Extension Header Identifier (EHI).</p> <p>NULL, is the default setting for ODU2 to 10G Ethernet or Pattern over GFP.</p> <p>LINear, selects the LINear configuration of Extension Header Identifier.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:CONF:EXI NULL SOUR:DATA:TEL:GFP:CONF:EXI? Returns NULL</pre>

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

Description	<p>This query returns the configuration of Extension Header Identifier (EHI).</p> <p>At *RST, this value is Null.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:GFP: CONFiguration:EXI?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Type></p>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character response data> element.</p> <p>Returns the configuration of Extension Header Identifier (EHI).</p> <p>NULL, is the default setting for ODU2 to 10G Ethernet or Pattern over GFP.</p> <p>LINear, LINear configuration of Extension Header Identifier is selected.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:CONF:EXI NULL</p> <p>SOUR:DATA:TEL:GFP:CONF:EXI? Returns NULL</p>

**:SOURCE[1..n]:DATA:TELEcom:GFP:
CONFIguration:UPI?**

Description	This query returns the configuration of User Payload Identifier (UPI). At *RST, this value is Framed 64B/66B ethernet.
Syntax	:SOURCE[1..n]:DATA:TELEcom:GFP: CONFIguration:UPI?
Parameter(s)	None
Response Syntax	<Type>

**:SOURCE[1..n]:DATA:TELEcom:GFP:
CONFiguration:UPI?**

Response(s)

Type:

The response data syntax for <Type> is defined as a <Character response data> element.

Returns the configuration of User Payload Identifier (UPI).

FETHERnet, FETHERNET is selected as the User Payload Identifier (UPI).

FPPP, FPPP is selected as the User Payload Identifier (UPI).

TFCHAnnel, TFCHANNEL is selected as the User Payload Identifier (UPI).

TFICON, TFICON is selected as the User Payload Identifier (UPI).

TESCON, TESCON is selected as the User Payload Identifier (UPI).

TGBEthernet, TGBETHERNET is selected as the User Payload Identifier (UPI).

FMAPOS, FMAPOS is selected as the User Payload Identifier (UPI).

TDVBASI, TDVBASI is selected as the User Payload Identifier (UPI).

**:SOURCE[1..n]:DATA:TELEcom:GFP:
CONFiguration:UPI?**

FRPR, FRPR is selected as the User Payload Identifier (UPI).

FFCBBW, FFCBBW is selected as the User Payload Identifier (UPI).

TAFCHannel, TAFCHannel is selected as the User Payload Identifier (UPI).

FMPLSUCAST, FMPLSUCAS is selected as the User Payload Identifier (UPI).

FMPLSMCAST, FMPLSMCAST is selected as the User Payload Identifier (UPI).

FISIS, FISIS is selected as the User Payload Identifier (UPI).

FIPV4, FIPV4 is selected as the User Payload Identifier (UPI).

FIPV6, FIPV6 is selected as the User Payload Identifier (UPI).

FDVBASI, FDVBAS is selected as the User Payload Identifier (UPI).

F64B66BETH, F64B66BETH is selected as the User Payload Identifier (UPI).

Example(s)

SOUR:DATA:TEL:GFP:CONF:UPI FETHernet
SOUR:DATA:TEL:GFP:CONF:UPI? Returns
FETHernet

**:SENSe[1..n]:DATA:TELEcom:GFP:
CONFIguration:UPI?**

Description	This query returns the configuration of User Payload Identifier (UPI). At *RST, this value is Framed 64B/66B ethernet.
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP: CONFIguration:UPI?
Parameter(s)	None
Response Syntax	<Type>

**:SENSe[1..n]:DATA:TELeCom:GFP:
CONFIguration:UPI?****Response(s)**

Type:

The response data syntax for <Type> is defined as a <Character response data> element.

Returns the configuration of User Payload Identifier (UPI).

FETHERnet, FETHERNET is selected as the User Payload Identifier (UPI).

FPPP, FPPP is selected as the User Payload Identifier (UPI).

TFCHAnnel, TFCHANNEL is selected as the User Payload Identifier (UPI).

TFICON, TFICON is selected as the User Payload Identifier (UPI).

TESCON, TESCON is selected as the User Payload Identifier (UPI).

TGBEthernet, TGBETHERNET is selected as the User Payload Identifier (UPI).

FMAPOS, FMAPOS is selected as the User Payload Identifier (UPI).

TDVBASI, TDVBASI is selected as the User Payload Identifier (UPI).

**:SENSe[1..n]:DATA:TELecom:GFP:
CONFIguration:UPI?**

FRPR, FRPR is selected as the User Payload Identifier (UPI).

FFCBBW, FFCBBW is selected as the User Payload Identifier (UPI).

TAFCHannel, TAFCHannel is selected as the User Payload Identifier (UPI).

FMPLSUCAST, FMPLSUCAS is selected as the User Payload Identifier (UPI).

FMPLSMCAST,FMPLSMCAST is selected as the User Payload Identifier (UPI).

FISIS, FISIS is selected as the User Payload Identifier (UPI).

FIPV4, FIPV4 is selected as the User Payload Identifier (UPI).

FIPV6, FIPV6 is selected as the User Payload Identifier (UPI).

FDVBASI, FDVBAS is selected as the User Payload Identifier (UPI).

F64B66BETH, F64B66BETH is selected as the User Payload Identifier (UPI).

Example(s)

SOUR:DATA:TEL:GFP:CONF:UPI FETHernet
SOUR:DATA:TEL:GFP:CONF:UPI? Returns
FETHernet

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:
FRAMe:TYPE**

Description	<p>This command sets the alarm for Loss of framed Delineation Type.</p> <p>At *RST, this value is LFD.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAMe:TYPE</pre>
Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: LFD</p> <p>Selects the alarm for Loss of framed Delineation Type.</p> <p>LFD, Loss of framed Delineation (LFD), indicates that GFP engine is out of synchronization.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:ALAR:FRAM:TYPE LFD SOUR:DATA:TEL:GFP:ALAR:FRAM:TYPE?</pre>
See Also	<pre>SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAMe:TYPE?</pre>

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:
FRAME:TYPE?**

Description	<p>This query returns the alarm for Loss of framed Delineation Type.</p> <p>At *RST, this value is LFD.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAME:TYPE?
Parameter(s)	None
Response Syntax	<Type>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character response data> element.</p> <p>Returns the alarm for Loss of framed Delineation Type.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ALAR:FRAM:TYPE LFD</p> <p>SOUR:DATA:TEL:GFP:ALAR:FRAM:TYPE?</p>
See Also	SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAME:TYPE

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:
FRAME**

Description	<p>This command sets the alarm for frames.</p> <p>At *RST, this value is LFD.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAME <wsp> <SET></pre>
Parameter(s)	<p>SET:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Sets the alarm for frames.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:ALAR:FRAM ON SOUR:DATA:TEL:GFP:ALAR:FRAM?</pre>
See Also	<pre>SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAME?</pre>

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:
FRAME?**

Description	This query returns the alarm for frames. At *RST, this value is LFD.
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAME?
Parameter(s)	None
Response Syntax	<Type>
Response(s)	Type: The response data syntax for <Type> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the alarm for frame.
Example(s)	SOUR:DATA:TEL:GFP:ALAR:FRAM ON SOUR:DATA:TEL:GFP:ALAR:FRAM?
See Also	SOURce[1..n]:DATA:TELEcom:GFP:ALARm: FRAME

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:MANual:TYPE****Description**

This command sets the type of manual error on Core header.

At *RST, this value is cHEC CORRectable.

At *RST, this value is cHEC CORRectable.

Syntax

:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:MANual:TYPE<wsp>
CORRectable|UCORRectable

:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:MANual:TYPE

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CORReactable UCORReactable.</p> <p>Selects the type of manual error.</p> <p>CORReactable, indicates that only one bit error has been detected on Core header.</p> <p>UCORReactable, indicates that two or more bit errors have been detected on Core header</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:MAN:TYPE CORReactable</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:MAN:TYPE?</p>
See Also	<p>SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:MANual:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:MANual:TYPE?**

Description	This query returns the type of manual error. At *RST, this value is cHEC CORRectable.
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Type>

:SOURce[1..n]:DATA:TELEcom:GFP:ERROr:FRAMe:MANual:TYPE?

Response(s)	Type: The response data syntax for <Type> is defined as a <Character response data> element. Returns the type of manual error. The possible values are: CORRectable UCORrectable. CORRectable, indicates that only one bit error has been detected on Core header. UCORrectable, indicates that two or more bit errors have been detected on Core header.
Example(s)	SOUR:DATA:TEL:GFP:ERR:FRAM:MAN:TYPE CORrectable SOUR:DATA:TEL:GFP:ERR:FRAM:MAN:TYPE?
See Also	SOURce[1..n]:DATA:TELEcom:GFP:ERROr:FRAMe:MANual:TYPE

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:AMOUnt**

Description	<p>This command sets the amount of manual errors to be generated.</p> <p>At *RST, this value is 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AMOUnt <wsp> MAXimum MINimum</pre>
Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Selects the amount of error.</p> <p>Choices are 1 to 50.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:ERR:FRAM:AMOUnt 40 SOUR:DATA:TEL:GFP:ERR:FRAM:AMOUnt?</pre>
See Also	<pre>SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAM e:AMOUnt?</pre>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:AMOUnt?**

Description	<p>This query returns the amount of manual errors to be generated.</p> <p>At *RST, this value is 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AMOUnt <wsp> MAXimum MINimum.</p>
Parameter(s)	<p>Amount: Amount: The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. Selects the amount of error. Choices are 1 to 50.</p>
Response Syntax	<p><Value></p>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:AMOut?**

Response(s)	Value: The response data syntax for <Value> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of manual errors to be generated.
Example(s)	SOUR:DATA:TEL:GFP:ERR:FRAM:AMOut 40 SOUR:DATA:TEL:GFP:ERR:FRAM:AMOut?
See Also	SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AMOut

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:INJect**

Description This command sets the manual or automated error injection.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:INJect<wsp>MAXimum | MINimum

Parameter(s) This parameter is optional.
If no token is specified, the current automated error is injected.

Example(s) SOUR:DATA:TEL:GFP:ERR:FRAM:INJect

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAME:AUTomated:TYPE****Description**

This command sets the automated error injection.

At *RST, this value is cHEC CORReactable.

Syntax

:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAME:AUTomated:TYPE<wsp>
CORReactable|UCORReactable

:SOURce[1..n]:DATA:TELEcom:GFP:ERROr: FRAMe:AUTomated:TYPE

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CORReactable UCORReactable.</p> <p>Selects the automated error type.</p> <p>CORReactable, selects a "walking 1" pattern to hit all applicable bits covered by the cHEC and PLI.</p> <p>UCORReactable, selects a "walking 11" pattern to hit all consecutive 2 bits applicable to the bits covered by the cHEC and PLI.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:TYPE CORReactable</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:TYPE?</p>
See Also	<p>SOURce[1..n]:DATA:TELEcom:GFP:ERROr: FRAMe:AUTomated:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:AUTomated:TYPE?**

Description	This query returns the automated error injection. At *RST, this value is cHEC CORRectable.
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Type>

:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:TYPE?

Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character response data> element.</p> <p>Returns the automated error injection.</p> <p>Possible values are: CORReactable UCORreactable.</p> <p>CORReactable, "walking 1" pattern is selected to hit all applicable bits covered by the cHEC and PLI.</p> <p>UCORreactable, "walking 11" pattern is selected to hit all consecutive 2 bits applicable to the bits covered by the cHEC and PLI.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:TYPE CORReactable</p> <p>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:TYPE?</p>
See Also	<p>SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAME:AUTomated:RATE**

Description	<p>This command sets the injection rate for the automated error.</p> <p>At *RST, this value is 1.0E-01.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAME:AUTomated:RATE<wsp> MAXimum MINimum</pre>
Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Selectstheinjectionratefortheautomatederror. Choices are 1.0E-09 through 1.0E-02.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:RATE SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:RATE?</pre>
See Also	<pre>SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAME:AUTomated:RATE?</pre>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERROr:
FRAMe:AUTomated:RATE?**

Description

This query returns the injection rate for the automated error.

At *RST, this value is 1.0E-01.

Syntax

:SOURce[1..n]:DATA:TELEcom:GFP:ERROr:
FRAMe:AUTomated:RATE<wsp>
MAXimum|MINimum

Parameter(s)

This parameter is optional.

If no token is specified, the injection rate for the automated error is 1.0E-01.

Response Syntax

<Rate>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:AUTomated:RATE?****Response(s)**

Rate:

The response data syntax for <TX> is defined as a <NR2 NUMERIC RESPONSE DATA> element.

Returns the injection rate for the automated error.

Example(s)

SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:RATE

SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:RATE?

See AlsoSOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:AUTomated:RATE

:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated

Description This command sets the automated error at a specified rate or continuously.

At *RST, this value is OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:AUTomated <wsp> <SET>

Parameter(s) SET:
The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.
Sets the automated error at a specified rate or continuously.

Example(s) SOUR:DATA:TEL:GFP:ERR:FRAM:AUT ON
SOUR:DATA:TEL:GFP:ERR:FRAM:AUT?

See Also SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:AUTomated?**

Description	<p>This query returns the automated error at a specified rate or continuously.</p> <p>At *RST, this value is OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated
Parameter(s)	None
Response Syntax	<SET>
Response(s)	<p>The response data syntax for <SET> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the automated error at a specified rate or continuously.</p>
Example(s)	SOUR:DATA:TEL:GFP:ERR:FRAM:AUT ON SOUR:DATA:TEL:GFP:ERR:FRAM:AUT?
See Also	SOURce[1..n]:DATA:TELEcom:GFP:ERRor:FRAM e:AUTomated

:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:CONTInuous

Description	<p>This command sets the automated error continuously.</p> <p>At *RST, this value is OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:CONTInuous <wsp> <SET></pre>
Parameter(s)	<p>SET:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Sets the automated error continuously.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:CONT ON SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:CONT?</pre>
See Also	<pre>SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:AU Tomated:CONTInuous?</pre>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:AUTomated:CONTInuous?**

Description	<p>This query returns the automated error continuously.</p> <p>At *RST, this value is OFF.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: FRAMe:AUTomated:CONTInuous?</code>
Parameter(s)	None
Response Syntax	<code><SET></code>
Response(s)	<p>SET:</p> <p>The response data syntax for <code><SET></code> is defined as a <code><NR1 NUMERIC RESPONSE DATA></code> element.</p> <p>Returns the automated error continuously.</p>
Example(s)	<code>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:CONT ON</code> <code>SOUR:DATA:TEL:GFP:ERR:FRAM:AUT:CONT?</code>
See Also	<code>SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:AUTomated:CONTInuous</code>

**:FETCh[1..n]:DATA:TELEcom:GFP:CONFig:
DELTA?**

Description	<p>This query returns the configuration of machine synchronization parameter.</p> <p>At *RST, this value is 1.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:GFP:CONFig: DELTA?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Value></p>
Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the configuration of machine synchronization parameter.</p>
Example(s)	<p>FETC:DATA:TEL:GFP:CONF:DELT?</p>

**:SENSe[1..n]:DATA:TELEcom:GFP:FRAMe:
MISMatch:COUNT?**

Description	<p>This query returns the count of frames with fields not matching the expected identifier.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:GFP:FRAMe: MISMatch:COUNT? <wsp>EXI UPI</pre>
Parameter(s)	<p>Count:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <numeric_value> elements for this parameter are: EXI UPI</p> <p>Selects the count of frames with fields not matching the expected identifier.</p> <p>EXI, selects the count of frames with fields not matching the expected identifier.</p> <p>UPI, selects the count of frames with fields not matching the expected identifier.</p>
Response Syntax	<pre><COUNT></pre>

:SENSe[1..n]:DATA:TELEcom:GFP:FRAME: MISMatch:COUNT?

Response(s)	COUNT: The response data syntax for <COUNT> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of frames with fields not matching the expected identifier.
Example(s)	FETC:DATA:TEL:GFP:ALAR:FRAM:HIST? LFD
See Also	FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:CURRent?

**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
FRAME:HISTory?**

Description	<p>This query returns the history status of LFD (Loss of Frame Delineation).</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAME:HISTory?<wsp> LFD </p>
Parameter(s)	<p>Count:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: LFD.</p> <p>Selects the history status of LFD.</p> <p>LFD, Indicates that GFP engine is out of synchronization.</p>
Response Syntax	<p><History></p>

:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:HISTory?

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <Character response data> element.</p> <p>Returns the count of the history status of Alarm analysis.</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>
Example(s)	<p>FETC:DATA:TEL:GFP:ALAR:FRAM:HIST? LFD</p>
See Also	<p>FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAMe:CURRent?</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
FRAME:CURRent?**

Description	<p>This query returns the current status of Alarm analysis.</p> <p>At *RST, this value is LFD (Loss of Frame Delineation).</p>
Syntax	FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAME:CURRent?<wsp> LFD
Parameter(s)	<p>Count:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: LFD.</p> <p>Selects the current status of Alarm analysis.</p> <p>LFD, generates a sufficient number of cHEC uncorrectable errors to avoid synchronization.</p>
Response Syntax	<current>

:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: FRAME:CURRent?

Response(s)	<p>current:</p> <p>The response data syntax for <current> is defined as a <Character response data> element.</p> <p>Returns the current status of Alarm analysis.</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>
Example(s)	<p>FETC:DATA:TEL:GFP:ALAR:FRAM:CURR? LFD</p>
See Also	<p>FETCh[1..n]:DATA:TELEcom:GFP:ALARm:FRAME: :HISTory?</p>

**:FETCh[1..n]:DATA:ELecom:GFP:ALARm:FRAME
:SEConds?**

Description	<p>This query returns the count of seconds in which alarm occurs.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:ELecom:GFP:ALARm:FRAME: SEConds?<wsp> LFD </pre>
Parameter(s)	<p>Count:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter is: LFD .</p> <p>Selects the count of seconds in which alarm occurs.</p> <p>LFD, generates a sufficient number of cHEC uncorrectable errors to avoid synchronization.</p>
Response Syntax	<pre><seconds></pre>

:FETCh[1..n]:DATA:ELecom:GFP:ALARm:FRAMe :SEConds?

Response(s)	seconds: The response data syntax for <seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the count of seconds in which alarm occurs.
Example(s)	FETC:DATA:TEL:GFP:ALAR:FRAM:SEC? LFD
See Also	FETCh[1..n]:DATA:TEL:GFP:ALAR:FRAM:CURR?

**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAME
:HISTory?**

Description	This query returns the history status of error analysis. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAME: HISTory? <wsp>CORRectable UCORrectable
Parameter(s)	<Count>
Response Syntax	<History>

:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAME :HISTory?

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <Character response data> element.</p> <p>Returns the history status of error analysis.</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>
Example(s)	<p>FETC:DATA:TEL:GFP:ERR:FRAM:HIST? CORREctable</p>
See Also	<p>FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAME: CURRent?</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe
:CURRent?**

Description	<p>This query returns the current status of error analysis.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe: CURRent? <wsp>CORReCtable UCORReCtable</code>
Parameter(s)	<p>Count:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CORReCtable UCORReCtable.</p> <p>Selects the current status of error.</p> <p>CORReCtable, selects the current status of CORReCtable error.</p> <p>UCORReCtable, selects the current status of UCORReCtable error.</p>
Response Syntax	<code><current></code>

:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAME :CURRent?

Response(s)	<p>current:</p> <p>The response data syntax for <current> is defined as a <Character response data> element.</p> <p>Returns the current status of error analysis.</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>
Example(s)	<p>FETC:DATA:TEL:GFP:ERR:FRAM:CURR? CORRectable</p>
See Also	<p>FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAME: HISTory?</p>

:FETCh[1..n]:DATA:TELEcom:GFP:ERROr:FRAMe:SECOnds?

Description	<p>This query returns the total number of seconds in which error occurs.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:GFP:ERROr:FRAMe:SECOnds?<wsp>CORReCtable UCORReCtable</code>
Parameter(s)	<p>Count:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CORReCtable UCORReCtable.</p> <p>Selects the total number of seconds in which error occurs.</p> <p>CORReCtable, selects the count of seconds in which CORReCtable error occurs.</p> <p>UCORReCtable, selects the count of seconds in which UCORReCtable error occurs.</p>
Response Syntax	<code><seconds></code>

**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
FRAME:SEConds?**

Response(s)	seconds: The response data syntax for <seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the total number of seconds in which error occurs.
Example(s)	FETC:DATA:TEL:GFP:ERR:FRAM:SEC? CORRectable
See Also	FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAME: RATE?

::FETCh[1..n]:DATA:TELEcom:GFP:ERROr:FRAME:COUNT?

Description	<p>This query returns the count of error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:GFP:ERROr:FRAME:COUNT?<wsp>CORReactable UCORReactable</code>
Parameter(s)	<p>Count:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CORReactable UCORReactable.</p> <p>Selects the count of error.</p> <p>CORReactable, selects the count of CORReactable error.</p> <p>UCORReactable, selects the count of CORReactable error.</p>
Response Syntax	<code><COUNT></code>

**::FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
FRAMe:COUNT?**

Response(s)	COUNT: The response data syntax for <COUNT> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the count of error.
Example(s)	FETC:DATA:TEL:GFP:ERR:FRAM:COUN? CORReactable
See Also	FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe: RATE?

**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe
:RATE?**

Description	<p>This query returns the calculations and displays the error rate.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>FETCh[1..n]:DATA:TELEcom:GFP:ERRor:FRAMe:RATE? <wsp>CORReactable UCORreactable</p>
Parameter(s)	<p>Count:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CORReactable UCORreactable.</p> <p>Selects the calculations and displays the error rate</p> <p>CORReactable, selects calculations and displays CORReactable error rate.</p> <p>UCORreactable, selects calculations and displays UCORreactable error rate.</p>
Response Syntax	<p><RATE></p>

:FETCh[1..n]:DATA:TELeom:GFP:ERRor:FRAMe :RATE?

Response(s)	<p>RATE:</p> <p>The response data syntax for <RATE> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the calculations and displays the error rate.</p>
Example(s)	<p>FETC:DATA:TEL:GFP:ERR:FRAM:RATE? CORReactable</p>
See Also	<p>FETCh[1..n]:DATA:TELeom:GFP:ERRor:FRAMe: COUNT?</p>

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

This command detects the presence of the expected payload FCS for the frames.

At *RST, the configuration is set to a device-dependent value.

Syntax

:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel:
CONFig:TYPE<wsp>CDFRames|CMRFrames

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

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CDFrames CMRFrames.</p> <p>Selects detecting the presence of the expected payload FCS for the frames.</p> <p>CDFrames, enables detecting the presence of the expected payload FCS for the Client Data frames.</p> <p>CMRFrames, enables detecting the presence of the expected payload FCS for the Client Management frames.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:CHAN:CONF:TYPE CDFR,ON</p> <p>SOUR:DATA:TEL:GFP:CHAN:CONF:TYPE?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:CHANnel:CONFig:TYPE?</p>

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

Description	<p>This query returns detecting the presence of the expected payload FCS for the frames.</p> <p>At *RST, the configuration is set to a device-dependent value.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:TYPE? <wsp>CDFRames CMRFrames</code>
Parameter(s)	<p>SET:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CDFRames CMRFrames.</p> <p>Selects detecting the presence of the expected payload FCS for the frames.</p> <p>CDFRames, enables detecting the presence of the expected payload FCS for the Client Data frames.</p> <p>CMRFrames, enables detecting the presence of the expected payload FCS for the Client Management frames.</p>
Response Syntax	<code><Type></code>

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

Response(s)	Type: The response data syntax for <Type> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Detects the presence of the expected payload FCS for the frames.
Example(s)	SOUR:DATA:TEL:GFP:CHAN:CONF:TYPE CDFR,ON SOUR:DATA:TEL:GFP:CHAN:CONF:TYPE?
See Also	SOURce[1..n]:TELEcom:GFP:CHANnel:CONFig:TYPE

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

Description	<p>This command sets the communication channel used for the signal reception.</p> <p>At *RST, the configuration is set 0.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:CID<wsp>MAXimum MINimum</pre>
Parameter(s)	<p>VALUE:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Selects the communication channel used for the signal reception.</p> <p>Choices are from 00000000 to 11111111 (0 to 255). Only available when EXI is set to Linear.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:CHAN:CONF:CID 230 SOUR:DATA:TEL:GFP:CHAN:CONF:CID?</pre>
See Also	<pre>SOURce[1..n]:TELEcom:GFP:CHANnel:CONFig: CID?</pre>

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

Description	<p>This query returns the communication channel used for the signal reception.</p> <p>At *RST, the configuration is set 0.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:CID? <wsp>MAXimum MINimum</p>
Parameter(s)	<p>This parameter is optional.</p> <p>If no token is specified, the configuration is 0.</p>
Response Syntax	<p><Type></p>

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

Response(s)	Type: The response data syntax for <Type> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the communication channel used for the signal reception.
Example(s)	SOUR:DATA:TEL:GFP:CHAN:CONF:CID 230 SOUR:DATA:TEL:GFP:CHAN:CONF:CID?
See Also	SOURce[1..n]:TELEcom:GFP:CHANnel:CONFig:CID

:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:TYPE

Description

This command sets the generation of Loss Of Client Signal by setting the UPI.

At *RST, the value is LOCS.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:  
CHANnel:TYPE<wsp>CLOCS|UDCMF|  
LOCCS|RDI|FDI|DCI
```

:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:TYPE

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: CLOCS UDCMF LOCCS RDI FDI DCI .</p> <p>Sets the generation of Loss Of Client Signal by setting the UPI.</p> <p>CLOCS, generates a LOCS by setting the UPI field to “0000 0001”.</p> <p>UDCMF, allows to set the User-defined UPI for the CMF value.</p> <p>LOCCS, generates a LOCCS by setting the UPI field to “0000 0010”.</p> <p>RDI, generates a client RDI by setting the UPI field to “0000 0101”.</p> <p>FDI, generates a client FDI by setting the UPI field to “0000 0100”.</p> <p>DCI, generates a client DCI by setting the UPI field to “0000 0011”.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN:TYPE CLOCS</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN:TYPE?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:ALARm:CHANnel:TYPE?</p>

:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:TYPE?

Description	<p>This query returns the generation of Loss Of Client Signal by setting the UPI.</p> <p>At *RST, the value is LOCS.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:TYPE?
Parameter(s)	None
Response Syntax	<Type>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character response data> element.</p> <p>Returns the generation of client management frames alarm.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN:TYPE CLOCS</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN:TYPE?</p>
See Also	SOURce[1..n]:TELEcom:GFP:ALARm:CHANnel:TYPE

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:PERiod?**

Description	This query returns the alarm period associated with the client management frames. At *RST, the value is 100 ms.
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:PERiod?<wsp>MAXimum MINimum
Parameter(s)	This parameter is optional. Choices are from 10 ms to 1200 ms.
Response Syntax	<Channel>

:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:PERiod?

Response(s)	<p>Channel:</p> <p>The response data syntax for <Channel> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the alarm period associated with the client management frames.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN:PER 100</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN:PER?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:ALARm:CHANnel: PERiod</p>

**::SOURce[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:UPI**

Description	<p>This command sets entering the client management frame UPI value when User Defined CMF is selected.</p> <p>At *RST, the value is 0000 0000.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel:UPI<wsp> <VALUE></pre>
Parameter(s)	<p>VALUE:</p> <p>The program data syntax for the parameter is defined as a <NON DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets entering the client management frame UPI value when User Defined CMF is selected.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:ALAR:CHAN:UPI #B00000000 SOUR:DATA:TEL:GFP:ALAR:CHAN:UPI?</pre>
See Also	<pre>SOURce[1..n]:TELEcom:GFP:ALARm:CHANnel:UPI?</pre>

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:UPI?**

Description	This query returns entering the client management frame UPI value when User Defined CMF is selected. At *RST, the value is 0000 0000.
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:UPI?
Parameter(s)	None
Response Syntax	<VALUE>

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:UPI?**

Response(s)	VALUE: The response data syntax for <VALUE> is defined as a <BINARY RESPONSE DATA> element. Returns entering the client management frame UPI value when User Defined CMF is selected.
Example(s)	SOUR:DATA:TEL:GFP:ALAR:CHAN:UPI #B00000000 SOUR:DATA:TEL:GFP:ALAR:CHAN:UPI?
See Also	SOURce[1..n]:TELEcom:GFP:ALARm:CHANnel: UPI

:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel

Description	<p>This command allows the generation of client management frames alarms.</p> <p>This command is not associated with any *RST value.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:CHANnel<wsp> <SET></pre>
Parameter(s)	<p>SET:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Sets the generation of client management frames alarm.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:ALAR:CHAN ON SOUR:DATA:TEL:GFP:ALAR:CHAN?</pre>
See Also	<pre>SOURce[1..n]:TELEcom:GFP:ALARm:CHANnel?</pre>

**:SOURce[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel?**

Description	<p>This command allows the generation of client management frames alarms.</p> <p>This command is not associated with any *RST value.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel?
Parameter(s)	None
Response Syntax	<VALUE>
Response(s)	<p>VALUE:</p> <p>The response data syntax for <VALUE> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the generation of client management frames alarm analysis.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ALAR:CHAN ON</p> <p>SOUR:DATA:TEL:GFP:ALAR:CHAN?</p>
See Also	SOURce[1..n]:TELEcom:GFP:ALARm:CHANnel

:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated:TYPE

Description

This command sets the error for automated injection mode.

At *RST, the value is tHEC Correctable.

Syntax

:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:CHAN
nel:AUTomated:TYPE<wsp>TCORrect|
TUCORrect|ECORrect|EUCORrect|PFCS

:SOURCE[1..n]:DATA:TELECOM:GFP:ERROR: CHANnel:AUTomated:TYPE

Parameter(s)	TYPE:
	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TCORrect TUCORrect ECORrect EUCORrect PFCS.</p> <p>Selects the error for automated injection mode</p> <p>TCORrect, generates a "Walking 1" pattern to hit all applicable bits covered by the tHEC, PTI, EXI and UPI.</p> <p>TUCORrect, generates a "Walking 11" pattern to hit all consecutive 2 bits applicable bits covered by the tHEC, PTI, EXI and UPI.</p> <p>ECORrect, generates a "Walking 1" pattern to hit all applicable bits covered by the eHEC, CID and Spare. Only available with Linear frames (EXI is set to llinear)</p> <p>EUCORrect, generates a "Walking 11" pattern to hit all consecutive 2 bits applicable bits covered by the eHEC, CID and Spare. Only available with Linear frames (EXI is set to llinear)</p> <p>PFCS, generates a "Walking 1" pattern to hit all 32 bits of the pFCS only. Only available when Client Data Frames FCS is enabled.</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:AUTomated:TYPE**

Example(s)	SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:TYPE TCORRect SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:TYPE?
See Also	SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:AU Tomated:TYPE?

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

Description	This query returns the error for automated injection mode. At *RST, the value is tHEC Correctable.
Syntax	:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR: CHANnel:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Type>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERROr:
CHANnel:AUTomated:TYPE?**

Response(s)

Type:

The response data syntax for <Type> is defined as a <Character response data> element.

Returns the error for automated injection mode.

Possible values are:

TCORrect|TUCORrect|ECORrect|EUCORrect|
PFCS.

TCORrect, "Walking 1" pattern is generated to hit all applicable bits covered by the tHEC, PTI, EXI and UPI.

TUCORrect, "Walking 11" pattern is generated to hit all consecutive 2 bits applicable bits covered by the tHEC, PTI, EXI and UPI.

ECORrect, "Walking 1" pattern is generated to hit all applicable bits covered by the eHEC, CID and Spare. Only available with Linear frames (EXI is set to lineair)

EUCORrect, "Walking 11" pattern is generated to hit all consecutive 2 bits applicable bits covered by the eHEC, CID and Spare. Only available with Linear frames (EXI is set to lineair)

PFCS, "Walking 1" pattern is generated to hit all 32 bits of the pFCS only. Only available when Client Data Frames FCS is enabled.

Example(s)

SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:TYPE
TCORrect

SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:TYPE?

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:AUTomated:TYPE?**

See Also

SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:AU
Tomated:TYPE

**:SOURce[1..n]:DATA:TELEcom:GFP:ERROR:
CHANnel:AUTomated:RATE**

Description	<p>This command sets the injection rate for automated error.</p> <p>At *RST, the value is 1.0E-01.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:GFP:ERROR: CHANnel:AUTomated:RATE <wsp> MAXimum MINimum</p>
Parameter(s)	<p>VALUE:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> element for this parameter are: MAXimum MINimum.</p> <p>Select the injection rate for automated error. Choices are from 9.9E-6 to 1.0E-1.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:RATE 1.2E-09</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:RATE?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:ERROR:CHANnel:AUTomated:RATE?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERROr:
CHANnel:AUTomated:RATE?**

Description	This query returns the injection rate for automated error. At *RST, the value is 1.0E-01.
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ERROr: CHANnel:AUTomated:RATE?<wsp>MAXimum MINimum
Parameter(s)	This parameter is optional. Choices are from 9.9E-6 to 1.0E-1.
Response Syntax	<RATE>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:AUTomated:RATE?**

Response(s)	<p>RATE:</p> <p>The response data syntax for <RATE> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for automated error.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:RATE 1.2E-09</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:RATE?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:AUTomated:RATE</p>

**:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:
CHANnel:AUTomated:CONTInuous**

Description	<p>When activated, generates the selected error for each frame to its theoretical maximum.</p> <p>At *RST, the value is set to OFF.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:GFP:ERROR:CHAN nel:AUTomated:CONTInuous <wsp> <set></pre>
Parameter(s)	<p>set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Sets the selected error for each frame to its theoretical maximum.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:CONT ON SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:CONT?</pre>
See Also	<pre>SOURCE[1..n]:TELEcom:GFP:ERROR:CHANnel:AU Tomated:CONTInuous?</pre>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:AUTomated:CONTInuous?**

Description	<p>This query returns the selected error for each frame to its theoretical maximum.</p> <p>At *RST, the value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<SET>
Response(s)	<p>SET:</p> <p>The response data syntax for <SET> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the selected error for each frame to its theoretical maximum.</p>
Example(s)	SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:CONT ON SOUR:DATA:TEL:GFP:ERR:CHAN:AUT:CONT?
See Also	SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:AU Tomated:CONTInuous

**:SOURce[1..n]:DATA:TELEcom:GFP:ERROr:
CHANnel:AUTomated**

Description	<p>This command sets the automated error. The On/Off button is used to activate/deactivate the selected automated error at the rate specified or continuously when continuous is enabled.</p> <p>At *RST, the value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ERROr: CHANnel:AUTomated <wsp> <set></pre>
Parameter(s)	<p>set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Sets the automated error at the rate specified or continuously when continuous is enabled.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT ON SOUR:DATA:TEL:GFP:ERR:CHAN:AUT?</pre>
See Also	<pre>SOURce[1..n]:TELEcom:GFP:ERROr:CHANnel:AU Tomated?</pre>

:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated?

Description	<p>This query returns the automated error at the rate specified or continuously when continuous is enabled.</p> <p>At *RST, the value is set to OFF.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AUTomated?</code>
Parameter(s)	None
Response Syntax	<code><SET></code>
Response(s)	<p>SET:</p> <p>The response data syntax for <code><SET></code> is defined as a <code><NR1 NUMERIC RESPONSE DATA></code> element.</p>
Example(s)	<code>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT ON</code> <code>SOUR:DATA:TEL:GFP:ERR:CHAN:AUT?</code>
See Also	<code>SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel: AUTomated</code>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:MANual:TYPE**

Description	<p>This command sets the error for manual injection mode.</p> <p>At *RST, the value is tHEC Correctable.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:MANual:TYPE<wsp>TCORrect TUCORrect ECORrect EUCORrect PFCS</pre>

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

Parameter(s)	TYPE:
	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TCORrect TUCORrect ECORrect EUCORrect PFCS.</p> <p>Selects the error for manual injection mode</p> <p>TCORrect, generates a "Walking 1" pattern to hit all applicable bits covered by the tHEC, PTI, EXI and UPI.</p> <p>TUCORrect, generates a "Walking 11" pattern to hit all consecutive 2 bits applicable bits covered by the tHEC, PTI, EXI and UPI.</p> <p>ECORrect, generates a "Walking 1" pattern to hit all applicable bits covered by the eHEC, CID and Spare. Only available with Linear frames (EXI is set to llinear)</p> <p>EUCORrect, generates a "Walking 11" pattern to hit all consecutive 2 bits applicable bits covered by the eHEC, CID and Spare. Only available with Linear frames (EXI is set to llinear)</p> <p>PFCS, generates a "Walking 1" pattern to hit all 32 bits of the pFCS only. Only available when Client Data Frames FCS is enabled.</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:MANual:TYPE****Example(s)**SOUR:DATA:TEL:GFP:ERR:CHAN:MAN:TYPE
TCORRect

SOUR:DATA:TEL:GFP:ERR:CHAN:MAN:TYPE?

See AlsoSOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:M
ANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:MANual:TYPE?**

Description	<p>This query returns the error for manual injection mode</p> <p>At *RST, the value is tHEC Correctable.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:MANual:TYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Type></p>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character response data> element.</p> <p>Returns the error for manual injection mode.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:MAN:TYPE TCORRect</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:MAN:TYPE?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:M ANual:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERROR:
CHANnel:AMOut**

Description	<p>This command sets the amount of manual error to be generated.</p> <p>At *RST, the value is 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:ERROR: CHANnel:AMOut <wsp> MAXimum MINimum</pre>
Parameter(s)	<p>VALUE:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Selects the amount of manual error to be generated.</p> <p>Choices are 1 through 50.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:ERR:CHAN:AMOut 50 SOUR:DATA:TEL:GFP:ERR:CHAN:AMOut?</pre>
See Also	<pre>SOURce[1..n]:TELEcom:GFP:ERROR:CHANnel: AMOut?</pre>

:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AMOut?

Description	This query returns the amount of manual error to be generated. At *RST, the value is 1.
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:AMOut?<wsp>MAXimum MINimum
Parameter(s)	This parameter is optional. Choices are 1 through 50.
Response Syntax	<RATE>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:AMOunt?**

Response(s)	<p>RATE:</p> <p>The response data syntax for <RATE> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of manual error to be generated.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:AMOunt 50</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:AMOunt?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel: AMOunt</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:MANual**

Description This command sets the manual error for the amount specified. The On/Off button is used to activate/deactivate the selected manual error for the amount specified. The On/Off button is automatically deactivated once the amount of error has been injected.

At *RST, the value is OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:MANual<wsp><set>

Parameter(s) set:
The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.
Sets the selected error for each frame to its theoretical maximum.

Example(s) SOUR:DATA:TEL:GFP:ERR:CHAN:MAN ON
SOUR:DATA:TEL:GFP:ERR:CHAN:MAN?

See Also SOURce[1..n]:TELEcom:GFP:ERRor:CHANnel:
MANual?

**:SOURce[1..n]:DATA:TELEcom:GFP:ERROR:
CHANnel:MANual?**

Description	<p>This query returns the manual error for the amount specified. The on/off button is automatically deactivated once the amount of error has been injected.</p> <p>At *RST, the value is OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:ERROR: CHANnel:MANual?
Parameter(s)	None
Response Syntax	<SET>

:SOURce[1..n]:DATA:TELEcom:GFP:ERROr: CHANnel:MANual?

Response(s)	<p>SET:</p> <p>The response data syntax for <SET> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the manual error for the amount specified. The on/off button is automatically deactivated once the amount of error has been injected.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:ERR:CHAN:MAN ON</p> <p>SOUR:DATA:TEL:GFP:ERR:CHAN:MAN?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:ERROr:CHANnel: MANual</p>

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

Description	This command sets the communication channel used for the signal transmission. *RST, the value is 0.
Syntax	:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:CID<wsp>MAXimum MINimum
Parameter(s)	This parameter is optional. Choices are from 00000000 to 11111111 (0 to 225).
Example(s)	FETC:DATA:TEL:GFP:CHAN:CONF:CID?

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

Description	This query returns the communication channel used for the signal transmission. *RST, the value is 0.
Syntax	:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel: CONFig:CID? <wsp>MAXimum MINimum
Parameter(s)	This parameter is optional. Choices are from 00000000 to 11111111 (0 to 225).
Response Syntax	<Type>
Response(s)	Type: The response data syntax for <Type> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the communication channel used for the signal transmission.
Example(s)	FETC:DATA:TEL:GFP:CHAN:CONF:CID?

**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:HISTory?****Description**

This query returns the history status of Alarm Analysis.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:HISTory?<wsp>LOCS|LOCCS|RDI
|FDI|DCI

**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:HISTory?**

Parameter(s)	<p>History:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOCS LOCCS RDI FDI DCI</p> <p>Returns the history status of Alarm Analysis.</p> <p>LOCS, generates a LOCS by setting the UPI field to “0000 0001”</p> <p>LOCCS, generates a LOCCS by setting the UPI field to “0000 0010”.</p> <p>RDI, generates a client RDI by setting the UPI field to “0000 0101”.</p> <p>FDI, generates a client FDI by setting the UPI field to “0000 0100”.</p> <p>DCI, generates a client DCI by setting the UPI field to “0000 0011”.</p>
Response Syntax	<History>

**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <Character response data> element.</p> <p>Returns the History status of Alarm Analysis.</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>
Example(s)	FETC:DATA:TEL:GFP:ALAR:CHAN:HIST? LOCS
See Also	FETCh[1..n]:TELEcom:GFP:ALARm:CHANnel: CURRent?

:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:CURRent?

Description This query returns the current status of Alarm Analysis.

At *RST, this value is device dependent.

Syntax FETCh[1..n]:DATA:TELEcom:GFP:ALARm:CHAN
nel:CURRent? <wsp>LOCS|LOCCS|RDI|FDI|
DCI

:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:CURRent?

Parameter(s)	<p>Current:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: LOCS LOCCS RDI FDI DCI.</p> <p>Returns the current status of Alarm Analysis.</p> <p>LOCS, generates a LOCS by setting the UPI field to “0000 0001”</p> <p>LOCCS, generates a LOCCS by setting the UPI field to “0000 0010”.</p> <p>RDI, generates a client RDI by setting the UPI field to “0000 0101”.</p> <p>FDI, generates a client FDI by setting the UPI field to “0000 0100”.</p> <p>DCI, generates a client DCI by setting the UPI field to “0000 0011”.</p>
Response Syntax	<Current>

:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:CURRent?

Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <Character response data> element.</p> <p>Returns the current status of Alarm Analysis.</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>
Example(s)	<p>FETC:DATA:TEL:GFP:ALAR:CHAN:SEC? LOCS</p>
See Also	<p>FETCh[1..n]:TELEcom:GFP:ALARm:CHANnel:CURRent?</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:SEConds?**

Description This query returns the total number of seconds in which alarm occurs.

*RST, the value is set to 0.

Syntax :FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:SEConds? <wsp>LOCS|CMF|LOCCS|
RDI|FDI|DCI

:FETCh[1..n]:DATA:TELEcom:GFP:ALARm: CHANnel:SECOnds?

Parameter(s)

History:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

LOCS|CMF|LOCCS|RDI|FDI|DCI.

Returns the total number of seconds in which alarm occurs.

LOCS, generates a LOCS by setting the UPI field to "0000 0001".

CMF, allows to set the User-defined UPI for the CMF value.

LOCCS, generates a LOCCS by setting the UPI field to "0000 0010".

RDI, generates a client RDI by setting the UPI field to "0000 0101".

FDI, generates a client FDI by setting the UPI field to "0000 0100".

DCI, generates a client DCI by setting the UPI field to "0000 0011".

Response Syntax

<Seconds>

**:FETCh[1..n]:DATA:TELEcom:GFP:ALARm:
CHANnel:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the total number of seconds in which alarm occurs.
Example(s)	FETC:DATA:TEL:GFP:ALAR:CHAN:SEC? LOCS
See Also	FETCh[1..n]:TELEcom:GFP:ALARm:CHANnel:CURRent?

**:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel:
MISMatch:COUNT?**

Description	<p>This query returns the number of frames with fields not matching the expected identifier.</p> <p>*RST, the value is set to 0.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel: MISMatch:COUNT? <wsp>PFI CID</p>
Parameter(s)	<p>COUNT:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter are: PFI CID.</p> <p>Returns the number of frames with fields not matching the expected identifier.</p> <p>PFI, Payload Frame Check Sequence Identifier allows overwriting the Payload FCS Indicator.</p> <p>CID, Channel Identifier, selects the communication channel used for the signal transmission. Choices are from 00000000 through 11111111 (0 to 255). The default value is 0 when EXI is set to Linear.</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:CHANnel:
MISMAtch:COUNT?****Response Syntax** <COUNT>**Response(s)** COUNT:
The response data syntax for <COUNT> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the number of frames with fields not matching the expected identifier.**Example(s)** FETC:DATA:TEL:GFP:CHAN:MISM:COUNT? PFI

**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:HISTory?**

Description

This query returns the history status of Error Analysis.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:HISTory?<wsp>TCORrect|
TUCORrect|ECORrect|EUCORrect|PFCS

**:FETCh[1..n]:DATA:TELEcom:GFP:ERROr:
CHANnel:HISTOrY?****Parameter(s)****TYPE:**

The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.

The allowed <numeric_value> elements for this parameter are: TCORrect | TUCORrect | ECORrect | EUCORrect | PFCS.

Returns the history status of errors available for both manual and automated injection modes.

TCORrect, indicates that only one bit error has been detected in the type header.

TUCORrect, indicates that two or more bit error have been detected in the type header.

ECORrect, indicates that only one bit error has been detected in the extension header. Only available with Linear Frames.

EUCORrect, indicates that two or more bit errors have been detected in the extension header. Only available with Linear Frames.

PFCS, indicates that at least one bit error has been detected in the payload.

Response Syntax

<History>

:FETCh[1..n]:DATA:TELEcom:GFP:ERROr: CHANnel:HISTory?

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <Character response data> element.</p> <p>Returns the History status of Error Analysis.</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>
Example(s)	<p>FETC:DATA:TEL:GFP:ERR:CHAN:HIST? TCORRect</p>
See Also	<p>FETCh[1..n]:TELEcom:GFP:ERROr:CHANnel:CURRent?</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:CURRent?****Description**

This query returns the current status of Error Analysis.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:CURRent? <wsp>TCORrect|TUCorrect
|ECORrect|EUCorrect|PFCS

**:FETCh[1..n]:DATA:TELEcom:GFP:ERROr:
CHANnel:CURRent?**

Parameter(s)

CURRent:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TCORrect | TUCORrect | ECORrect | EUCORrect | PFCS.

Returns the current status of errors available for both manual and automated injection modes.

TCORrect, indicates that only one bit error has been detected in the type header.

TUCORrect, indicates that two or more bit error have been detected in the type header.

ECORrect, indicates that only one bit error has been detected in the extension header. Only available with Linear Frames.

EUCORrect, indicates that two or more bit errors have been detected in the extension header. Only available with Linear Frames.

PFCS, indicates that at least one bit error has been detected in the payload.

Response Syntax

<Current>

**:FETCh[1..n]:DATA:TELecom:GFP:ERROr:
CHANnel:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <Character response data> element.</p> <p>Returns the current status of Error Analysis.</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>
Example(s)	<p>FETC:DATA:TEL:GFP:ERR:CHAN:CURR? TCORRect</p>
See Also	<p>FETCh[1..n]:TELecom:GFP:ERROr:CHANnel: HISTory?</p>

:FETCh[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:SEConds?

Description This query returns the total number of seconds in which error occurs.

*RST, the value is set to 0.

Syntax :FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:SEConds? <wsp>TCORrect|TUCORrect
|ECORrect|EUCORrect|PFCS

:FETCh[1..n]:DATA:TELEcom:GFP:ERROr: CHANnel:SECOnds?

Parameter(s)	<p>SECOnds:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TCORrect TUCORrect ECORrect EUCORrect PFCS.</p> <p>Selects the number of seconds in which error occurs.</p> <p>TCORrect, indicates that only one bit error has been detected in the type header.</p> <p>TUCORrect, indicates that two or more bit error have been detected in the type header.</p> <p>ECORrect, indicates that only one bit error has been detected in the extension header. Only available with Linear Frames.</p> <p>EUCORrect, indicates that two or more bit errors have been detected in the extension header. Only available with Linear Frames.</p> <p>PFCS, indicates that at least one bit error has been detected in the payload.</p>
Response Syntax	<Seconds>

**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the total number of seconds in which error occurs.
Example(s)	FETC:DATA:TEL:GFP:ERR:CHAN:SEC? TCORRect
See Also	FETCh[1..n]:TELEcom:GFP:ERRor:CHANnel:CURRent?

**:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:Count?**

Description This query returns the no of occurrences of error.

*RST, the value is set to 0.

Syntax :FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:Count? <wsp>TCORrect|TUCORrect|
ECORrect|EUCORrect|PFCS

**:FETCh[1..n]:DATA:TELEcom:GFP:ERROr:
CHANnel:Count?**

Parameter(s)

Count:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TCORrect | TUCORrect | ECORrect | EUCORrect | PFCS.

Selects the no of occurrences of error.

TCORrect, indicates that only one bit error has been detected in the type header.

TUCORrect, indicates that two or more bit error have been detected in the type header.

ECORrect, indicates that only one bit error has been detected in the extension header. Only available with Linear Frames.

EUCORrect, indicates that two or more bit errors have been detected in the extension header. Only available with Linear Frames.

PFCS, indicates that at least one bit error has been detected in the payload.

Response Syntax

<COUNT>

**:FETCh[1..n]:DATA:TELEcom:GFP:ERROr:
CHANnel:Count?**

Response(s)	COUNT: The response data syntax for <COUNT> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the no of occurrences of error.
Example(s)	FETC:DATA:TEL:GFP:ERR:CHAN:COUN? TCORRect
See Also	FETCh[1..n]:TELEcom:GFP:ERROr:CHANnel: RATE?

:FETCh[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:Rate?

Description

This query returns the error injection rate.

*RST, the value is set to 0.

Syntax

:FETCh[1..n]:DATA:TELEcom:GFP:ERRor:
CHANnel:Rate?<wsp>TCORrect|TUCorrect|
ECORrect|EUCorrect|PFCS

**:FETCh[1..n]:DATA:TELEcom:GFP:ERROr:
CHANnel:Rate?****Parameter(s)**

Rate:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TCORrect | TUCORrect | ECORrect | EUCORrect | PFCS.

Selects the error rate.

TCORrect, indicates that only one bit error has been detected in the type header.

TUCORrect, indicates that two or more bit error have been detected in the type header.

ECORrect, indicates that only one bit error has been detected in the extension header. Only available with Linear Frames.

EUCORrect, indicates that two or more bit errors have been detected in the extension header. Only available with Linear Frames.

PFCS, indicates that at least one bit error has been detected in the payload.

Response Syntax

<RATE>

:FETCh[1..n]:DATA:TELEcom:GFP:ERRor: CHANnel:Rate?

Response(s)	<p>RATE:</p> <p>The response data syntax for <RATE> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the error rate.</p>
Example(s)	<p>FETC:DATA:TEL:GFP:ERR:CHAN:RATE? TCORRect</p>
See Also	<p>FETCh[1..n]:TELEcom:GFP:ERRor:CHANnel: COUNT?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:PTI****Description**

This command displays the type of GFP client frame.

At *RST, the value is 000.

Syntax

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:PTI<wsp>DFRames|MFRames, <Pti>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI

Parameter(s)	<p>Frames:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames.</p> <p>Displays the type of GFP client frame.</p> <p>DFRmes, displays the type of GFP for the DFRmes.</p> <p>MFRames, displays the type of GFP for the MFRames.</p> <p>Pti:</p> <p>The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>Allows overwriting the Payload Type Identifier for the selected frame type.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:OH:THE:PTI MFR,#B00</p> <p>SOUR:DATA:TEL:GFP:OH:THE:PTI?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:OH:THEader:PTI?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:PTI?**

Description	<p>This query displays the type of GFP client frame.</p> <p>At *RST, the value is 000.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader :PTI? <wsp>DFRames MFRames</pre>
Parameter(s)	<p>Frames:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames.</p> <p>Displays the type of GFP client frame.</p> <p>DFRmes, displays the type of GFP for the DFRmes.</p> <p>MFRames, displays the type of GFP for the MFRames.</p>
Response Syntax	<pre><Pti></pre>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PTI?

Response(s)	Pti: The response data syntax for <Pti> is defined as a <BINARY RESPONSE DATA> element. Displays the type of GFP client frame.
Example(s)	SOUR:DATA:TEL:GFP:OH:THE:PTI MFR,#B00 SOUR:DATA:TEL:GFP:OH:THE:PTI?
See Also	SOURce[1..n]:TELEcom:GFP:OH:THEader:PTI

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:PFI****Description**

This command displays the Payload FCS Indicator.

At *RST, the value is 0.

Syntax

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:PFI<wsp>DFRames|MFRames, <Pfi>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI

Parameter(s)	<p>Frames:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames.</p> <p>Displays the Payload FCS Indicator</p> <p>DFRmes, displays the Payload FCS Indicator of DFRmes.</p> <p>MFRames, displays the Payload FCS Indicator of MFRames.</p> <p>Pfi:</p> <p>The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Allows overwriting the Payload FCS Indicator.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:OH:THE:PFI MFR,#B00</p> <p>SOUR:DATA:TEL:GFP:OH:THE:PFI?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:OH:THEader:PFI?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:PFI?**

Description	<p>This query displays the Payload FCS Indicator.</p> <p>At *RST, the value is 0.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader :PFI? <wsp>DFRames MFRames</pre>
Parameter(s)	<p>Frames:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames.</p> <p>Displays payload FCS indicator.</p> <p>DFRmes, selects the display payload FCS indicator of DFRmes.</p> <p>MFRames, selects the display payload FCS indicator of MFRames.</p>
Response Syntax	<pre><Pfi></pre>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:PFI?

Response(s)	Pfi: The response data syntax for <Pfi> is defined as a <BINARY RESPONSE DATA> element. Displays the Payload FCS Indicator.
Example(s)	SOUR:DATA:TEL:GFP:OH:THE:PFI MFR,#B00 SOUR:DATA:TEL:GFP:OH:THE:PFI?
See Also	SOURce[1..n]:TELEcom:GFP:OH:THEader:PFI

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:EXI****Description**

This command displays the Extension Header Identifier.

At *RST, the value is 0000.

Syntax

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:EXI<wsp>DFRames|MFRames, <Exi>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI

Parameter(s)	<p>Frames:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames.</p> <p>Displays Extension Header identifier.</p> <p>DFRmes, displays Extension Header identifier of DFRmes.</p> <p>MFRames, displays Extension Header identifier of MFRames.</p> <p>Exi:</p> <p>The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>Allows overwriting the Extension Header Identifier.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:OH:THE:EXI MFR,#B0000</p> <p>SOUR:DATA:TEL:GFP:OH:THE:EXI?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:OH:THEader:EXI?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:EXI?**

Description	<p>This query displays the Extension Header Identifier.</p> <p>At *RST, the value is 0000.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader :EXI? <wsp>DFRames MFRames</pre>
Parameter(s)	<p>Frames:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames.</p> <p>Display Extension Header identifier.</p> <p>DFRmes, displays Extension Header identifier of DFRmes.</p> <p>MFRames, displays Extension Header identifier of MFRames.</p>
Response Syntax	<pre><Exi></pre>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:EXI?

Response(s)	Exi: The response data syntax for <Exi> is defined as a <BINARY RESPONSE DATA> element. Displays the Extension Header Identifier.
Example(s)	SOUR:DATA:TEL:GFP:OH:THE:EXI MFR,#B0000 SOUR:DATA:TEL:GFP:OH:THE:EXI?
See Also	SOURce[1..n]:TELEcom:GFP:OH:THEader:EXI

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:UPI****Description**

This command displays the User Payload Identifier.

At *RST, the value is 00010011.

Syntax

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader
:UPI<wsp>DFRames|MFRames, <Upi>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI

Parameter(s)	<p>Frames:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames.</p> <p>Displays User Payload identifier.</p> <p>DFRmes, displays User Payload identifier of DFRmes.</p> <p>MFRames, displays User Payload identifier of MFRames.</p> <p>Upi:</p> <p>The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Allows overwriting the User Payload Identifier.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:OH:THE:UPI MFR,#B0000</p> <p>SOUR:DATA:TEL:GFP:OH:THE:UPI?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:OH:THEader:UPI?</p>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader :UPI?

Description	<p>This query displays the User Payload Identifier.</p> <p>At *RST, the value is 00010011.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader :UPI?<wsp>DFRames MFRames
Parameter(s)	<p>Frames:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames.</p> <p>Displays User Payload identifier.</p> <p>DFRmes, displays User Payload identifier of DFRmes.</p> <p>MFRames, displays User Payload identifier of MFRames.</p>
Response Syntax	<Upi>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:THEader:UPI?

Response(s)	Upi: The response data syntax for <Upi> is defined as a <BINARY RESPONSE DATA> element. Displays the User Payload identifier.
Example(s)	SOUR:DATA:TEL:GFP:OH:THE:UPI MFR,#B0000 SOUR:DATA:TEL:GFP:OH:THE:UPI?
See Also	SOURce[1..n]:TELEcom:GFP:OH:THEader:UPI

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHeader
:CID**

Description	<p>This command displays the communication channel used by the signal. The possible values are 00000000 through 11111111 (0 to 255).</p> <p>At *RST, the value is OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHeader :CID<wsp>DFRames MFRames, <CID></pre>

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHeader
:CID**

Parameter(s)	<p>Frames:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames.</p> <p>Displays the communication channel used by the signal.</p> <p>DFRmes, displays the communication channel used by the signal of DFRmes.</p> <p>MFRames, displays the communication channel used by the signal of MFRames.</p> <p>CID:</p> <p>The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Allows to overwrite the communication channel used for the signal transmission set from GFP Channel TX.</p> <p>Choices are from 00000000 through 11111111 (0 to 255). The default setting is 00000000.</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:OH:EHE:CID DFR, #B00000000</p> <p>SOUR:DATA:TEL:GFP:OH:EHE:CID?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:OH:EHeader:CID?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHeader
:CID?**

Description	<p>This query displays the communication channel used by the signal. The possible values are 00000000 through 11111111 (0 to 255).</p> <p>At *RST, the value is OFF.</p>
Syntax	<p>SOURce[1..n]:DATA:TELEcom:GFP:OH:EHeader: CID? <wsp>DFRmes MFRmes</p>
Parameter(s)	<p>Frames:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>The allowed <numeric_value> elements for this parameter are: DFRmes MFRmes.</p> <p>Displays of the communication channel used by the signal.</p> <p>DFRmes, displays the communication channel used by the signal of DFRmes.</p> <p>MFRmes, displays the communication channel used by the signal of MFRmes.</p>
Response Syntax	<p><CID></p>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader :CID?

Response(s)	<p>CID:</p> <p>The response data syntax for <CID> is defined as a <BINARY RESPONSE DATA> element.</p> <p>Returns the communication channel used by the signal. The possible values are 00000000 through 11111111 (0 to 255).</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:OH:EHE:CID DFR, #B00000000</p> <p>SOUR:DATA:TEL:GFP:OH:EHE:CID?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:OH:EHEader:CID</p>

**:SOURCE[1..n]:DATA:TELEcom:GFP:OH:EHeader
:SPARe****Description**

Displays the Extension header Spare field. The possible values are 00000000 through 11111111 (0 to 255).

At *RST, the value is 0.

Syntax

:SOURCE[1..n]:DATA:TELEcom:GFP:OH:EHeader
:SPARe<wsp>DFRames|MFRames, <Spare>

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader
:SPARe**

Parameter(s)	<p>Frames:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: DFRames MFRames.</p> <p>Display the Extension header Spare field.</p> <p>DFRmes, displays the Extension header Spare field of DFRmes.</p> <p>MFRames, displays the Extension header Spare field of MFRames.</p> <p>Spare:</p> <p>The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>Allows to set the extension header Spare field.</p> <p>Choices are from 00000000 through 11111111 (0 to 255).</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:OH:EHE:SPAR DFR, #B00000000</p> <p>SOUR:DATA:TEL:GFP:OH:EHE:SPAR?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:OH:EHEader: SPARe?</p>

**:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHeader
:SPARe?**

Description	Displays the Extension header Spare field. The possible values are 00000000 through 11111111 (0 to 255). At *RST, the value is 0.
Syntax	:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHeader :SPARe?
Parameter(s)	None
Response Syntax	<Pti>

:SOURce[1..n]:DATA:TELEcom:GFP:OH:EHEader :SPARe?

Response(s)	<p>Pti:</p> <p>The response data syntax for <Pti> is defined as a <BINARY RESPONSE DATA> element.</p> <p>Returns the display of the Extension header Spare field. The possible values are 00000000 through 11111111 (0 to 255).</p>
Example(s)	<p>SOUR:DATA:TEL:GFP:OH:EHE:SPAR DFR, #B00000000</p> <p>SOUR:DATA:TEL:GFP:OH:EHE:SPAR?</p>
See Also	<p>SOURce[1..n]:TELEcom:GFP:OH:EHEader: SPARe</p>

**:FETCh[1..n]:DATA:TELEcom:GFP:OH:
DFRames?****Description**

This query returns the selection of the Client Data Frames.

At *RST, the value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:GFP:OH:DFRames?
<wsp>PLI|CHEC|PTI|PFI|EXI|UPI|THEC|CID|
SPARe|EHEC

**:FETCh[1..n]:DATA:TELEcom:GFP:OH:
DFRames?**

Parameter(s)	Dframes: The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PLI CHEC PTI PFI EXI UPI THEC CID SPARe EHEC. SelectstheClientDataFrames. PLI, indicates the number of octets in the GFP payload area. CHEC, indicates the CRC-16 error control code that protects the integrity of the contents of the core header by enabling both single-bit error correction and multi-bit error detection. PTI, displays the type of GFP client frame. PFI, displays the payload Frame Check Sequence Indicator. EXI, displays the Extension Header Identifier. UPI, displays the User Payload Identifier. THEC, indicated the CRC-16 error control code that protects the integrity of the contents of the field by enabling both sinle-bit error correction and multi-bit error detection. CID, displays the communication channel used by the signal. SPARe, displays the extension header Spare field. EHEC, indicates the CRC-16 error control code that protects the integrity of the contents of the extension header by enabling both single-bit error correction and multi-bit error detection.
---------------------	---

**:FETCh[1..n]:DATA:TELEcom:GFP:OH:
DFRames?****Response Syntax** <Dframes>**Response(s)**

Dframes:

The response data syntax for <Dframes> is defined as a <BINARY RESPONSE DATA> element.

Returns the selection of the Client Data Frames.

Example(s)FETC:DATA:TEL:GFP:OH:DFRAM? PLI

:FETCh[1..n]:DATA:TELEcom:GFP:OH: MFRames?

Description

This query returns the selection of the Client Management Frames.

At *RST, the value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:GFP:OH:MFRames?
<wsp>PLI|CHEC|PTI|PFI|EXI|UPI|THEC|CID|
SPARe|EHEC

:FETCh[1..n]:DATA:TELEcom:GFP:OH: MFRames?

Parameter(s)	
	Mframes:
	The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
	The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PLI CHEC PTI PFI EXI UPI THEC CID SPARe EHEC.
	Selects the Client Management Frames.
	PLI, indicates the number of octets in the GFP payload area.
	CHEC, indicates the CRC-16 error control code that protects the integrity of the contents of the core header by enabling both single-bit error correction and multi-bit error detection.
	PTI, displays the type of GFP client frame.
	PFI, displays the payload Frame Check Sequence Indicator.
	EXI, displays the Extension Header Identifier.
	UPI, displays the User Payload Identifier.
	THEC, indicated the CRc-16 error control code that protects the integrity of the contents of the field by enabling both sinle-bit error correction and multi-bit error detection.
	CID, displays the communication channel used by the signal.
	SPARe, displays the extension header Spare field.
	EHEC, indicates the CRC-16 error control code that protects the integrity of the contents of the extension header by enabling both single-bit error correction and multi-bit error detection.

:FETCh[1..n]:DATA:TELEcom:GFP:OH: MFRames?

Response Syntax <Mframes>

Response(s) Mframes:
The response data syntax for <Mframes> is defined as a <BINARY RESPONSE DATA> element.
Returns the selection of the Client Management Frames.

Example(s) FETC:DATA:TEL:GFP:OH:MFRAM? PLI

**:FETCh[1..n]:DATA:TELEcom:GFP:OH:
RPTiframes?****Description**

This query returns the selection of the Reserved PTI Frames.

At *RST, the value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:GFP:OH:
RPTiframes?<wsp>PLI|CHEC|PTI|PFI|EXI|UPI
|THEC|CID|SPARe|EHEC

:FETCh[1..n]:DATA:TELEcom:GFP:OH: RPTiframes?

Parameter(s)

Rptiframes:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: PLI|CHEC|PTI|PFI|EXI|UPI|THEC|CID|SPARe|EHEC.

Selects the Reserved PTI Frames.

PLI, indicates the number of octets in the GFP payload area.

CHEC, indicates the CRC-16 error control code that protects the integrity of the contents of the core header by enabling both single-bit error correction and multi-bit error detection.

PTI, displays the type of GFP client frame.

PFI, displays the payload Frame Check Sequence Indicator.

EXI, displays the Extension Header Identifier.

UPI, displays the User Payload Identifier.

THEC, indicated the CRC-16 error control code that protects the integrity of the contents of the field by enabling both sinle-bit error correction and multi-bit error detection.

CID, displays the communication channel used by the signal.

SPARe, displays the extension header Spare field.

EHEC, indicates the CRC-16 error control code that protects the integrity of the contents of the extension header by enabling both single-bit error correction and multi-bit error detection.

**:FETCh[1..n]:DATA:TELEcom:GFP:OH:
RPTiframes?****Response Syntax** <Rptiframes>**Response(s)**

Rptiframes:

The response data syntax for <Rptiframes> is defined as a <BINARY RESPONSE DATA> element.

Returns the selection of the Reserved PTI Frames.

Example(s)FETC:DATA:TEL:GFP:OH:RPT? PLI

:SOURce[1..n]:DATA:TELEcom:GFP:CLient: PAYLoad:SIZE

Description	<p>This command sets the size of the data structure.</p> <p>At *RST, the value is set to 1500 bytes.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:GFP:CLient: PAYLoad:SIZE<wsp>MAXimum MINimum</pre>
Parameter(s)	<p>VALUE:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Selects the size of the data structure.</p> <p>Choices are 1 through 65523 bytes for Linear Extension with pFCS.</p> <p>1 through 65527 bytes for Linear Extension without pFCS or Null Extension with pFCS.</p> <p>1 through 65531 bytes for Null Extension without pFCS.</p>
Example(s)	<pre>SOUR:DATA:TEL:GFP:CLI:PAYL:SIZE 100 SOUR:DATA:TEL:GFP:CLI:PAYL:SIZE?</pre>
See Also	<pre>SOURce[1..n]:TELEcom:GFP:CLient:PAYLoad: SIZE?</pre>

**:SOURCE[1..n]:DATA:TELEcom:GFP:CLient:
PAYLoad:SIZE?**

Description	This query returns the size of the data structure. At *RST, the value is set to 1500 bytes.
Syntax	:SOURCE[1..n]:DATA:TELEcom:GFP:CLient: PAYLoad:SIZE? <wsp> MAXimum MINimum
Parameter(s)	This parameter is optional. Choices are 1 through 65523 bytes for Linear Extension with pFCS. 1 through 65527 bytes for Linear Extension without pFCS or Null Extension with pFCS. 1 through 65531 bytes for Null Extension without pFCS.
Response Syntax	<Size>

:SOURce[1..n]:DATA:TELEcom:GFP:CLient: PAYLoad:SIZE?

Response(s)	Size: The response data syntax for <Size> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the size of the data structure.
Example(s)	SOUR:DATA:TEL:GFP:CLI:PAYL:SIZE 100 SOUR:DATA:TEL:GFP:CLI:PAYL:SIZE?
See Also	SOURce[1..n]:TELEcom:GFP:CLient:PAYLoad:SIZE

**:SOURCE[1..n]:DATA:TELEcom:GFP:CLient:
INTerface**

Description	<p>This command allows the Ethernet interface type selection.</p> <p>At *RST, the value is set to Electrical.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:GFP:CLient: INTerface<wsp>OPTical ELEctrical</pre>
Parameter(s)	<p>Interface:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPTical ELEctrical.</p> <p>AllowstheEthernetinterfacetypeselection. OPTical,selectstheOpticalinterface. ELEctrical , selects the Electrical interface.</p>

FC Overclocked 1F

SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:FEC

Description	<p>This command enables or disables the Forward Error Correction (FEC) for non standard rates OTU1f/2f of the transmitter.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: FEC<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Enables or disables the Forward Error Correction (FEC) for the transmitter.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:F:FEC ON * SOUR:DATA:TEL:OTN:OTU1:F:FEC? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:FEC?</pre>

**SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
F:FEC?**

Description	<p>This query returns the status of Forward Error Correction (FEC) for non standard rates OTU1f/2f of the transmitter.</p> <p>At *RST, this value is set to ON.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:FEC?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of Forward Error Correction (FEC) for the transmitter.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OTU1:F:FEC ON</p> <p>* SOUR:DATA:TEL:OTN:OTU1:F:FEC? Returns 1</p>
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:FEC

**SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
F:FEC**

Description	<p>This command enables or disables the Forward Error Correction (FEC) for non standard rates OTU1f/2f of the receiver.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:FEC<wsp><Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Enables or disables the Forward Error Correction (FEC) for the transmitter.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OTU1:F:FEC ON</p> <p>* SENS:DATA:TEL:OTN:OTU1:F:FEC? Returns 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:FEC?</p>

**SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:
F:FEC?**

Description	<p>This query returns the status of Forward Error Correction (FEC) for non standard rates OTU1f/2f of the receiver.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<code>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:FEC?</code>
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of Forward Error Correction (FEC) for the receiver.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OTU1:F:FEC ON</p> <p>* SENS:DATA:TEL:OTN:OTU1:F:FEC? Returns 1</p>
See Also	* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:FEC?

SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:SCRambler

Description	<p>This command enables or disables the scrambler for non standard rates OTU1f/2f of the transmitter.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SCRambler<wsp>Set</p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Enables or disables the scrambler of the transmitter.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OTU1:F:SCR ON</p> <p>* SOUR:DATA:TEL:OTN:OTU1:F:SCR? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SCRambler?</p>

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

Description	<p>This query returns the status of scrambler for non standard rates OTU1f/2f of the transmitter.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: SCRambler?</pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Returns the status of the scrambler for the transmitter.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:F:SCR ON * SOUR:DATA:TEL:OTN:OTU1:F:SCR? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:SCRambler</pre>

SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:SCRambler

Description	<p>This command enables or disables the scrambler for non standard rates OTU1f/2f of the receiver.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: SCRambler<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Enables or disables the scrambler of the transmitter.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:OTU1:F:SCR ON * SENS:DATA:TEL:OTN:OTU1:F:SCR? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: SCRambler?</pre>

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

Description	<p>This query returns the status of the scrambler for non standard rates OTU1f/2f of the receiver.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SCRambler?</pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Returns the status of the scrambler for the receiver.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:OTU1:F:SCR ON * SENS:DATA:TEL:OTN:OTU1:F:SCR? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SCRambler</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SM:SAPI:B16

Description	<p>This command sets the injected message for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:SM:SAPI:B16<wsp><Message></pre>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the selected message.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:F:SM:SAPI:B16 "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:OTU1:F:SM:SAPI:B16? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:SM:SAPI:B16?</pre>

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

Description	This query returns the injected message for non standard rates OTU1f/2f. At *RST, this value is set to "EXFO OTU SAPI".
Syntax	: SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:SM:SAPI:B16?
Parameter(s)	None
Response Syntax	<Message>

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SM:SAPI:B16?

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the selected message.
Example(s)	* SOUR:DATA:TEL:OTN:OTU1:F:SM:SAPI:B16 "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:OTU1:F:SM:SAPI:B16? Returns "EXFO OTU SAPI"
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SM:SAPI:B16

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

Description	<p>This command sets the injected message for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:SM:DAPI:B16<wsp><Message></pre>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the selected message.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:F:SM:DAPI:B16 "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:OTU1:F:SM:DAPI:B16? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n] :F:SM:DAPI:B16?</pre>

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

Description	This query returns the injected message for non standard rates OTU1f/2f. At *RST, this value is set to "EXFO OTU DAPI".
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n] :F:SM:DAPI:B16?
Parameter(s)	None
Response Syntax	<Message>

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SM:DAPI:B16?

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the selected message.
Example(s)	* SOUR:DATA:TEL:OTN:OTU1:F:SM:DAPI:B16 "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:OTU1:F:SM:DAPI:B16? Returns "EXFO OTU DAPI"
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SM:DAPI:B16

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: SM:OPSPec:B32

Description	<p>This command sets the injected message for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F :SM:OPSPec:B32 <wsp> <Message></pre>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the selected message.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:F:SM:OPSP:B32 "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:OTU1:F:SM:OPSP:B32? Returns "EXFO OTU OPERATOR SPECIFIC"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:SM:OPSPec:B32?</pre>

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

Description	<p>This query returns the injected message for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:SM:OPSPec:B32?
Parameter(s)	None
Response Syntax	<Message>

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SM:OPSPec:B32?

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the selected message.
Example(s)	* SOUR:DATA:TEL:OTN:OTU1:F:SM:OPSP:B32 "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:OTU1:F:SM:OPSP:B32? Returns "EXFO OTU OPERATOR SPECIFIC"
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SM:OPSPec:B32

**:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:
SM:OVERwrite:ENABled**

Description	<p>This command enables or disables the SM Overwrite feature for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: SM:OVERwrite:ENABled<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the SM Overwrite feature.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OTU1:F:SM:OVER:ENAB ON * SOUR:DATA:TEL:OTN:OTU1:F:SM:OVER:ENAB? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]: F:SM:OVERwrite:ENABled?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SM:OVERwrite:ENABLEd?

Description	<p>This query returns the status of the SM Overwrite feature for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SM:OVERwrite:ENABLEd?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the SM Overwrite feature.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OTU1:F:SM:OVER:ENAB ON</p> <p>* SOUR:DATA:TEL:OTN:OTU1:F:SM:OVER:ENAB? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:SM:OVERwrite:ENABLEd</p>

:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:SAPI:EXPEcted

Description	<p>This command sets the expected message for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:SAPI:EXPEcted <wsp> <Message></pre>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the expected message.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:OTU1:F:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:OTU1:F:TTI:SAPI:EXP "EXFO OTU SAPI" * SENS:DATA:TEL:OTN:OTU1:F:TTI:SAPI:EXP? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:TIM * SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:SAPI:EXPEcted?</pre>

:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:SAPI:EXPEcted?

Description	This query returns the expected message for non standard rates OTU1f/2f. At *RST, this value is set to "EXFO OTU SAPI".
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:SAPI:EXPEcted?
Parameter(s)	None
Response Syntax	<Message>

**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F:
TTI:SAPI:EXPeCted?****Response(s)**

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the expected message.

Example(s)

* SENS:DATA:TEL:OTN:OTU1:F:TTI:TIM SAPI,ON

* SENS:DATA:TEL:OTN:OTU1:F:TTI:SAPI:EXP
"EXFO OTU SAPI"

* SENS:DATA:TEL:OTN:OTU1:F:TTI:SAPI:EXP?

Returns "EXFO OTU SAPI"

See Also

* SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F:
TTI:TIM

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

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

Description	<p>This command sets expected message for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:SAPI:EXPeCted <wsp> <Message></pre>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the expected message.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:OTU1:F:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:OTU1:F:TTI:DAPI:EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:OTU1:F:TTI:DAPI:EXP? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:TIM * SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:DAPI:EXPeCted?</pre>

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

Description	This query returns expected message for non standard rates OTU1f/2f. At *RST, this value is set to "EXFO OTU DAPI".
Syntax	SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:DAPI:EXPeCted?
Parameter(s)	None
Response Syntax	<Message>

:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:DAPI:EXPeCted?

Response(s)

Message:

The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.

Returns the expected message.

Example(s)

* SENS:DATA:TEL:OTN:OTU1:F:TTI:TIM DAPI,ON

* SENS:DATA:TEL:OTN:OTU1:F:TTI:DAPI:EXP

"EXFO OTU DAPI"

* SENS:DATA:TEL:OTN:OTU1:F:TTI:DAPI:EXP?

Returns "EXFO OTU DAPI"

See Also

* SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F:
TTI:TIM

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

**:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:
TTI:SAPI:B16?**

Description	<p>This query returns the received message for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:SAPI:B16?
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OTU1:F:TTI:SAPI:B16?</p> <p>Returns "EXFO OTU SAPI"</p>

:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:DAPI:B16?

Description	<p>This query returns the received message for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:DAPI:B16?</pre>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message.</p>
Example(s)	<pre>* FETC:DATA:TEL:OTN:OTU1:F:TTI:DAPI:B16?</pre> <p>Returns "EXFO OTU DAPI"</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:
TTI:OPSPec:B32?**

Description	<p>This query returns the received message for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:OPSPec:B32?
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OTU1:F:TTI:OPSP:B32?</p> <p>Returns "EXFO OTU OPERATOR SPECIFIC"</p>

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

Description This command enables or disables the state of Trace Identifier Mismatch (TIM) for non standard rates OTU1f/2f.

At *RST, this value is set to ON.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F:
TTI:TIM<wsp>SAPI|DAPI,<Set>

Parameter(s) Etim:
The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SAPI|DAPI.
Enables or disables the Trace Identifier Mismatch (TIM).
SAPI, selects the SAPI which allows editing of the Source Access Point Identifier (SAPI) message to be generated.
DAPI, selects the DAPI which allows editing of the Destination Access Point Identifier (DAPI) message to be generated.

**:SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F:
TTI:TIM****Set:**

The program data syntax for <Set> is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability.

ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the Trace Identifier Mismatch (TIM).

Example(s)

* SENS:DATA:TEL:OTN:OTU1:F:TTI:TIM SAPI,ON

* SENS:DATA:TEL:OTN:OTU1:F:TTI:TIM? SAPI

Returns 1

See Also

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

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

Description	<p>This query returns status of Trace Identifier Mismatch (TIM) for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]:F: TTI:TIM? <wsp>SAPI DAPI</p>
Parameter(s)	<p>Etim:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SAPI DAPI.</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p> <p>SAPI, selects the SAPI which allows editing of the Source Access Point Identifier (SAPI) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the Destination Access Point Identifier (DAPI) message to be generated.</p>
Response Syntax	<p><Set></p>

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

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Trace Identifier Mismatch (TIM).
Example(s)	* SENS:DATA:TEL:OTN:OTU1:F:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:OTU1:F:TTI:TIM? SAPI Returns 1
See Also	* SENSe[1..n]:DATA:TELecom:OTN:OTU[1..n]:F: TTI:TIM

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:TTI:OVERwrite:ENABLEd**

Description	<p>This command enables or disables the Trail Trace Identifier (TTI) Overwrite feature for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:TTI:OVERwrite:ENABLEd <wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Trail Trace Identifier (TTI) Overwrite feature.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:TTI:OVER:ENAB ON</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:TTI:OVER: ENAB? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:TTI:OVERwrite:ENABLEd?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:TTI:OVERwrite:ENABLEd?**

Description	This query returns the status of the Trail Trace Identifier (TTI) Overwrite feature for non standard rates OTU1f/2f. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:TTI:OVERwrite:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:TTI:OVERwrite:ENABLEd?

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Trail Trace Identifier (TTI) Overwrite feature.
Example(s)	* SOUR:DATA:TEL:OTN:ODU1:F:TTI:OVER:ENAB ON * SOUR:DATA:TEL:OTN:ODU1:F:TTI:OVER: ENAB? Returns 1
See Also	* SOURce[1..n]:DATA:TELecom:OTN:ODU[1..n]: F:TTI:OVERwrite:ENABLEd

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:FTFL:INDication****Description**

This command sets the FTFL Fault Indication message to be generated for non standard rates OTU1f/2f.

At *RST, this value is set to NFAult.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
:FTFL:INDication<wsp>FORWARD|BACKward,
NFAult|SFAil|SDEGrade|REServed

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

Parameter(s)

Ftfl:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWARD | BACKWARD.

Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).

FORWARD, sets the Forward configuration.

BACKWARD, sets the Backward configuration.

Indication:

The program data syntax for the second parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
NFAULT | SFAIL | SDEGRADE | RESERVED.

Sets the FTFL Fault Indication message to be generated.

NFAULT, selects the No Fault (NFAULT) as fault indication message.

SFAIL, selects the Signal Fail (SFAIL) as fault indication message.

SDEGRADE, selects the Signal Degraded (SDEGRADE) as fault indication message.

RESERVED, selects the Reserved as fault indication message.

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

Example(s) * SOUR:DATA:TEL:OTN:ODU1:F:FTFL:IND
FORW,SFA
* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:IND?
FORW Returns SFAIL

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:FTFL:INDication?

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

Description	<p>This query returns the FTFL Fault Indication message to be generated for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to NFAult.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:FTFL:INDication? <wsp>FORWARD BACKward</p>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter arF: 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>
Response Syntax	<p><Indication></p>

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

Response(s)	<p>Indication:</p> <p>The response data syntax for <Indication> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:IND FORW,SFA</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:IND? FORW Returns SFAIL</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:INDication</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:CODE

Description This command sets the FTFL Fault Indication code to be generated for non standard rates OTU1f/2f.

At *RST, this value is set to #H00.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
:FTFL:CODE<wsp>FORWard|BACKward,
<Code>

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

Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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> <p>Code:</p> <p>The program data syntax for the second parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the FTFL Fault Indication Code to be generated.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:CODE FORW,#H01</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:CODE? FORW Returns #H01</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:CODE?</p>

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

Description	<p>This query returns the FTFL Fault Indication code to be generated for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to #H00.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: :FTFL:CODE? <wsp>FORWard BACKward</p>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter arF: 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>
Response Syntax	<p><Code></p>

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

Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the selection of the Fault Type Fault Location (FTFL) fault indication code to be generated.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:CODE FORW,#H01</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:CODE? FORW Returns #H01</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:CODE</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:IDENtifier

Description

This command sets the FTFL Operator Identifier to be generated for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:  
:FTFL:IDENtifier<wsp>FORWard|BACKward,  
<Identifier>
```

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

Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>
	<p>Identifier:</p> <p>The program data syntax for the second parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the operator identifier to be generated.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:IDEN FORW,"exfo"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:IDEN? FORW Returns "exfo"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:IDENtifier?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
FTFL:IDENTifier?**

Description This query returns the FTFL Operator Identifier to be generated for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
FTFL:IDENTifier? <wsp>FORWARD|BACKWARD

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWARD|BACKWARD.
Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).
FORWARD, sets the Forward configuration.
BACKWARD, sets the Backward configuration.

Response Syntax <Identifier>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:FTFL:IDENTifier?**

Response(s)	Identifier: The response data syntax for <Identifier> is defined as a <STRING RESPONSE DATA> element. 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.
Example(s)	* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:IDEN FORW,"exfo" * SOUR:DATA:TEL:OTN:ODU1:F:FTFL:IDEN? FORW Returns "exfo"
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:IDENTifier

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:OPSPec

Description

This command sets the FTFL Operator Specific to be generated for non standard rates OTU1f/2f.

At *RST, this value is set to "EXFO ODU OPERATOR SPECIFIC".

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
:FTFL:OPSPec <wsp>FORWARD | BACKward,
<Specific>

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

Parameter(s)**Ftfl:**

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWARD|BACKWARD.

Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).

FORWARD, sets the Forward configuration.

BACKWARD, sets the Backward configuration.

Specific:

The program data syntax for the second parameter is defined as a <STRING PROGRAM DATA> element.

Sets the operator specific to be generated.

Example(s)

```
* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:OPSP
FORW,"exfo"
```

```
* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:OPSP?
FORW Returns "exfo"
```

See Also

```
* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:FTFL:OPSPec?
```

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:OPSPec?

Description	<p>This query returns the FTFL Operator Specific to be generated for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to "EXFO ODU OPERATOR SPECIFIC".</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:FTFL:OPSPec?<wsp>FORWard BACKward</p>
Parameter(s)	<p>Ftfl:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter arF: 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>
Response Syntax	<p><Specific></p>

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

Response(s)	<p>Specific:</p> <p>The response data syntax for <Specific> is defined as a <STRING RESPONSE DATA> 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>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:OPSP FORW,"exfo"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:OPSP? FORW Returns "exfo"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:OPSPec</p>

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

Description This query returns the FTFL Fault Indication message to be generated for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
FTFL:INDication?<wsp>FORWARD|BACKWARD

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWARD|BACKWARD.
Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).
FORWARD, sets the Forward configuration.
BACKWARD, sets the Backward configuration.

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:FTFL:INDication?****Response Syntax** <Indication>**Response(s)**

Indication:

The response data syntax for <Indication> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the FTFL Fault Indication message to be generated.

NFAULT, No Fault (NFAULT) is selected as FTFL Fault Indication message.

SFAIL, Signal Fail (SFAIL) is selected as FTFL Fault Indication message.

SDEGRADE, Signal Degraded (SDEGRADE) is selected as FTFL Fault Indication message.

RESERVED, Reserved is selected as FTFL Fault Indication message.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:F:FTFL:IND? FORW
Returns the FTFL Fault Indication message to be generated.

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:CODE?

Description This query returns the FTFL Fault Indication code to be generated for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
FTFL:CODE?<wsp>FORWARD|BACKward

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWARD|BACKward.
Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).
FORWARD, sets the Forward configuration.
BACKward, sets the Backward configuration.

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:FTFL:CODE?****Response Syntax** <Code>**Response(s)**

Code:

The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the FTFL Fault Indication Code to be generated.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:F:FTFL:CODE?
FORW Returns the FTFL Fault Indication code to be generated.

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:IDENtifier?

Description This query returns the FTFL Operator Identifier to be generated for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
FTFL:IDENtifier? <wsp>FORWard|BACKward

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWard|BACKward.
Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).
FORWard, sets the Forward configuration.
BACKward, sets the Backward configuration.

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:
F:FTFL:IDENTifier?****Response Syntax** <Identifier>**Response(s)**

Identifier:

The response data syntax for <Identifier> is defined as a <STRING RESPONSE DATA> element.

Returns the value of the FTFL Operator Identifier (bytes 1 to 9 for forward, byte 129 to 137 for backward) to be generated.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:F:FTFL:IDEN?
FORW Returns the FTFL Operator Identifier to be generated.

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:FTFL:SPECific?**

Description This query returns the FTFL Operator Specific to be generated for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
FTFL:SPECific? <wsp>FORWARD|BACKward

Parameter(s) Fttl:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
FORWARD|BACKward.
Sets the configuration of the forward and backward ODU Fault Type Fault Location (FTFL).
FORWARD, sets the Forward configuration.
BACKward, sets the Backward configuration.

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:
F:FTFL:SPECific?****Response Syntax** <Specific>**Response(s)**

Specific:

The response data syntax for <Specific> is defined as a <STRING RESPONSE DATA> element.

Returns the value of FTFL Operator Specific (bytes 10 to 127 for forward, byte 138 to 255 for backward) to be generated.

Example(s)

* SENS:DATA:TEL:OTN:ODU1:F:FTFL:SPEC?
FORW Returns the FTFL Operator Specific to be generated.

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:OVERwrite:ENABLEd

Description	<p>This command enables or disables the Fault Type Fault Location (FTFL) Overwrite feature for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:FTFL:OVERwrite:ENABLEd<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Fault Type Fault Location (FTFL) Overwrite feature.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:OVER:ENAB ON * SOUR:DATA:TEL:OTN:ODU1:F:FTFL:OVER:ENAB? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:FTFL:OVERwrite:ENABLEd?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:FTFL:OVERwrite:ENABLEd?**

Description	This query returns the status of the Fault Type Fault Location (FTFL) Overwrite feature for non standard rates OTU1f/2f. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:OVERwrite:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:FTFL:OVERwrite:ENABLEd?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of the Fault Type Fault Location (FTFL) Overwrite feature.

Example(s)

* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:OVER:
ENAB ON

* SOUR:DATA:TEL:OTN:ODU1:F:FTFL:OVER:
ENAB? Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:
F:FTFL:OVERwrite:ENABLEd

:SOURCE[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE

Description	<p>This command sets the injected payload signal type to be generated for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to PRBStest.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F :PTYPE<wsp>EXPerimental ASYNchronous BISYNch ATM GFP1 VCONcate BSTiming BSNTiming ODUMux RFSTandard RPRopriet NULLtest PRBStest NAVailable</pre>
Parameter(s)	<p>Payload:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>EXPerimental ASYNchronous BISYNch ATM GFP1 VCONcate BSTiming BSNTiming ODUMux RFSTandard RPRopriet NULLtest PRBStest NAVailable.</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE

Sets the expected payload signal type to be generated.

EXPerimental, selects the payload type as Experimental.

ASYNchronous, selects the payload type as Asynchronous.

BISYNch, selects the payload type as Bit Synchronous (BISYNch).

ATM, selects the payload type as Asynchronous Transfer Mode (ATM).

GFP1, selects the payload type as Generic Framing Procedure (GFP).

VCONcate, selects the payload type as Virtual Concatenation (VCONcate).

BSTiming, selects the payload type as Bit Stream Timing (BSTiming).

BSNTiming, selects the payload type as Bit Stream No Timing (BSNTiming).

ODUMux, selects the payload type as Optical Data Unit Mux (ODUMux).

RFSTandard, selects the payload type as Reserved Future Standardization (RFSTandard).

RPRopriet, selects the payload type as Reserved Proprietary (RPRopriet).

NULLtest, selects the payload type as NULL Test.

PRBStest, selects the payload type as Pseudo Random Bit Sequence Test (PRBStest).

NAVailable, selects the payload type as Not Available.

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

Example(s)	* SOUR:DATA:TEL:OTN:OPU1:F:PTYP EXP * SOUR:DATA:TEL:OTN:OPU1:F:PTYP? Returns EXPERIMENTAL
Note	The Code field is automatically updated when the injected payload type is changed and vice versa.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPe?

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
F:PTYPE?**

Description	<p>This query returns the injected payload signal type to be generated for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to PRBStest.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Payload></p>
Response(s)	<p>Payload:</p> <p>The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the injected payload signal type to be generated</p> <p>EXPERIMENTAL, Experimental is selected as payload type.</p> <p>ASYNCHRONOUS, Asynchronous is selected as payload type.</p> <p>BISYNCH, Bit Synchronous (BISYNCH) is selected as payload type.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
F:PTYPE?**

ATM, Asynchronous Transfer Mode (ATM) is selected as payload type.

GFP1, Generic Framing Procedure (GFP) is selected as payload type.

VCONCATENATE, Virtual Concatenation (VCONCATENATE) is selected as payload type.

BSTIMING, Bit Stream Timing (BSTIMING) is selected as payload type.

BSNTIMING, Bit Stream No Timing (BSNTIMING) is selected as payload type.

ODUMUX, Optical Data Unit (ODU) Mux is selected as payload type.

RFSTANDARD, Reserved Future Standardization (RFSTANDARD) is selected as payload type.

RPROPRIET, Reserved Proprietary (RPROPRIET) is selected as payload type.

NULLTEST, NULL Test is selected as payload type.

PRBSTEST, Pseudo Random Bit Sequence Test (PRBSTEST) is selected as payload type.

NAVAILABLE, Not available is selected as payload type.

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE?

Example(s)

* SOUR:DATA:TEL:OTN:OPU1:F:PTYP EXP

* SOUR:DATA:TEL:OTN:OPU1:F:PTYP?

Returns EXPERIMENTAL

Note

The Code field is automatically updated when the injected payload type is changed and vice versa.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
F:PTYPE?

**:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F:
PTYPF:RECEived?**

Description	This query returns the received payload signal type to be generated for non standard rates OTU1f/2f. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F: :PTYPF:RECEived?
Parameter(s)	None
Response Syntax	<Payload>

:FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:F: PTYPF:RECeived?

Response(s)

Payload:

The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the received payload type.

EXPERIMENTAL, Experimental is retrieved.

ASYNCHRONOUS, Asynchronous is retrieved.

BISYNCH, Bit Synchronous (BISYNCH) is retrieved.

ATM, Asynchronous Transfer Mode (ATM) is retrieved.

GFP1, Generic Framing Procedure (GFP) is retrieved.

VCONCATENATE, Virtual Concatenation (VCONCATENATE) is retrieved.

BSTIMING, Bit Stream Timing (BSTIMING) is retrieved.

BSNTIMING, Bit Stream No Timing (BSNTIMING) is retrieved.

ODUMUX, Optical Data Unit (ODU) Mux is retrieved.

RFSTANDARD, Reserved Future Standardization (RFSTANDARD) is retrieved.

RPROPRIET, Reserved Proprietary (RPROPRIET) is retrieved.

NULLTEST, NULL Test is retrieved.

**:FETCh[1..n]:DATA:TELecom:OTN:OPU[1..n]:F:
PTYPF:RECeived?**

PRBSTEST, Pseudo Random Bit Sequence Test (PRBSTEST) is retrieved.

NAVAILABLE, Not available is retrieved.

Example(s)

* FETC:DATA:TEL:OTN:OPU1:F:PTYP:REC?

Returns the received payload type.

:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPE

Description	<p>This command sets the expected payload signal type to be generated for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to PRBStest.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F: :PTYPE<wsp>EXPerimental ASYNchronous BISYNch ATM GFP1 VCONcate BSTiming BSNTiming ODUMux RFSTandard RPRopriet NULLtest PRBStest NAVailable</pre>
Parameter(s)	<p>Payload:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>EXPerimental ASYNchronous BISYNch ATM GFP1 VCONcate BSTiming BSNTiming ODUMux RFSTandard RPRopriet NULLtest PRBStest NAVailable.</pre>

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

Sets the expected payload signal type to be generated.

EXPerimental, selects the payload type as Experimental.

ASYNchronous, selects the payload type as Asynchronous.

BISYNch, selects the payload type as Bit Synchronous (BISYNch).

ATM, selects the payload type as Asynchronous Transfer Mode (ATM).

GFP1, selects the payload type as Generic Framing Procedure (GFP).

VCONcate, selects the payload type as Virtual Concatenation (VCONcate).

BSTiming, selects the payload type as Bit Stream Timing (BSTiming).

BSNTiming, selects the payload type as Bit Stream No Timing (BSNTiming).

ODUMux, selects the payload type as Optical Data Unit Mux (ODUMux).

RFSTandard, selects the payload type as Reserved Future Standardization (RFSTandard).

RPRopriet, selects the payload type as Reserved Proprietary (RPRopriet).

NULLtest, selects the payload type as NULL Test.

PRBStest, selects the payload type as Pseudo Random Bit Sequence Test (PRBStest).

NAVailable, selects the payload type as Not Available.

:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]: F:PTYPe

Example(s)

* SENS:DATA:TEL:OTN:OPU1:F:PTYP EXP
* SENS:DATA:TEL:OTN:OPU1:F:PTYP?
Returns EXPERIMENTAL

See Also

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

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

Description	<p>This query returns the expected payload signal type to be generated for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to PRBStest.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F:PTYPE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Payload></p>
Response(s)	<p>Payload:</p> <p>The response data syntax for <Payload> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the injected payload signal type to be generated.</p> <p>EXPERIMENTAL, Experimental is selected as payload type.</p> <p>ASYNCHRONOUS, Asynchronous is selected as payload type.</p> <p>BISYNCH, Bit Synchronous (BITS) is selected as payload type.</p>

:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]: F:PTYPe?

ATM, Asynchronous Transfer Mode (ATM) is selected as payload type.

GFP, Generic Framing Procedure (GFP) is selected as payload type.

VCONCATENATE, Virtual Concatenation (VCONCATENATE) is selected as payload type.

BSTIMING, Bit Stream Timing (BSTIMING) is selected as payload type.

BSNTIMING, Bit Stream No Timing (BSNTIMING) is selected as payload type.

ODUMUX, Optical Data Unit (ODU) Mux is selected as payload type.

RFSTANDARD, Reserved Future Standardization (RFSTANDARD) is selected as payload type.

RPROPRIET, Reserved Proprietary (RPROPRIET) is selected as payload type.

NULLTEST, NULL Test is selected as payload type.

PRBSTEST, Pseudo Random Bit Sequence Test (PRBSTEST) is selected as payload type.

NAVAILABLE, Not available is selected as payload type.

Example(s)

* SENS:DATA:TEL:OTN:OPU1:F:PTYP EXP

* SENS:DATA:TEL:OTN:OPU1:F:PTYP?

Returns EXPERIMENTAL

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

See Also

* SENSE[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F:
PTYPE

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPF:OVERwrite:ENABLEd

Description This command enables or disables the Payload Type Overwrite feature for non standard rates OTU1f/2f.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
F:PTYPF:OVERwrite:ENABLEd <wsp> <Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
F:PTYPF:OVERwrite:ENABLEd**

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Payload Type Overwrite feature.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OPU1:F:PTYP:OVER: ENAB ON</p> <p>* SOUR:DATA:TEL:OTN:OPU1:F:PTYP:OVER: ENAB? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPF:OVERwrite:ENABLEd?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPF:OVERwrite:ENABLEd?

Description	This query returns the status of the Payload Type Overwrite feature for non standard rates OTU1f/2f. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPF:OVERwrite:ENABLEd?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
F:PTYPF:OVERwrite:ENABled?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element. Returns the status of the Payload Type Overwrite feature.
Example(s)	* SOUR:DATA:TEL:OTN:OPU1:F:PTYP:OVER: ENAB ON * SOUR:DATA:TEL:OTN:OPU1:F:PTYP:OVER: ENAB? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PTYPF:OVERwrite:ENABled

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PCODE

Description	<p>This command sets the corresponding injected payload type as hexadecimal code for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to #H03.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F :PCODE<wsp> <Code></pre>
Parameter(s)	<p>Code:</p> <p>The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the corresponding injected payload type.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:OPU1:F:PCOD #H00 * SOUR:DATA:TEL:OTN:OPU1:F:PCOD? Returns #H00</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PCODE?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
F:PCODE?**

Description	This query returns the corresponding injected payload type as hexadecimal code for non standard rates OTU1f/2f. At *RST, this value is set to #H03.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PCODE?
Parameter(s)	None
Response Syntax	<Code>

:SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PCODE?

Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the corresponding injected payload type as hexadecimal code.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:OPU1:F:PCOD #H00</p> <p>* SOUR:DATA:TEL:OTN:OPU1:F:PCOD?</p> <p>Returns #H00</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:OPU[1..n]: F:PCODE</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:
F:PCode:RECeived?**

Description	<p>This query returns the corresponding received payload type as hexadecimal code for non standard rates OTU1f/2f.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F: PCode:RECeived?</code>
Parameter(s)	None
Response Syntax	<code><Code></code>
Response(s)	<p>Code:</p> <p>The response data syntax for <code><Code></code> is defined as a <code><HEXADECIMAL NUMERIC RESPONSE DATA></code> element.</p> <p>Returns the received payload code.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:OPU1:F:PCOD:REC?</p> <p>Returns the received payload code.</p>

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

Description	<p>This command sets the corresponding expected payload type as hexadecimal code for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to #H03.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F: PCODE<wsp> <Code></p>
Parameter(s)	<p>Code:</p> <p>The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Sets the corresponding expected payload type as hexadecimal code.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OPU1:F:PCOD #H00 * SENS:DATA:TEL:OTN:OPU1:F:PCOD? Returns #H00</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F: PCODE?</p>

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

Description	This query returns the corresponding expected payload type as hexadecimal code for non standard rates OTU1f/2f. At *RST, this value is set to #H03.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F:PCODE?
Parameter(s)	None
Response Syntax	<Code>

:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]: F:PCODE?

Response(s)	<p>Code:</p> <p>The response data syntax for <Code> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the corresponding expected payload type as hexadecimal code.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OPU1:F:PCOD #H00</p> <p>* SENS:DATA:TEL:OTN:OPU1:F:PCOD?</p> <p>Returns #H00</p>
See Also	<p>* SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:F:PCODE</p>

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

Description	<p>This command enables or disables the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis for non standard rates OTU1f/2f.</p> <p>At *RST this value is set to OFF.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F :PLM<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:OPU1:F:PLM ON * SENS:DATA:TEL:OTN:OPU1:F:PLM? Returns 1</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F: PLM?</pre>

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

Description	This query returns the status of the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis for non standard rates OTU1f/2f. At *RST this value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:OPU[1..n]:F:PLM?
Parameter(s)	None
Response Syntax	<Set>

**:SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:
F:PLM?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the Optical Payload Unit-Payload Label Mismatch (OPU-PLM) alarm analysis.
Example(s)	* SENS:DATA:TEL:OTN:OPU1:F:PLM ON * SENS:DATA:TEL:OTN:OPU1:F:PLM? Returns 1
See Also	* SENSe[1..n]:DATA:TELecom:OTN:OPU[1..n]:F:PLM

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F:TYPE

Description This command selects the Optical Transport Unit (OTU) alarm type for non standard rates OTU1f/2f.

At *RST, this value is set to O AIS.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:TYPE<wsp>O AIS|O BDI|LOF2|
OO F1|LOM|OOM|OBlae|OIAE

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
O AIS|O BDI|LOF2|OO F1|LOM|OOM|OBlae|OIAE.
Selects the Optical Transport Unit (OTU) alarm type.
O AIS, selects Optical Transport Unit - Alarm Indication Signal (O AIS) which generates the polynomial numbers 11 (PN-11) over all OTU frame bits including FAS and MFAS continuously.

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F:TYPE

OBDI, selects 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.

LOF2, selects Loss of Frame (LOF) which generates the errors in all the FAS bits continuously.

OOF1, selects Out of Frame (OOF) which generates the errors in all the FAS bits for 5 consecutive OTU frames repetitively.

LOM, selects Loss of Multiframe (LOM) which generates the errors in multiframe numbers for all the OTU frames continuously.

OOM, selects Out of Multiframe (OOM) which generates the errors in multiframe numbers for 5 consecutive OTU frames repetitively.

OBlae, selects 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.

OIAE, selects 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.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OTU1:F:TYPE OAIS

* SOUR:DATA:TEL:OTN:ALAR:OTU1:F:TYPE?
Returns OAIS

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:TYPE**

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:TYPE?

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F:TYPE?

Description	<p>This query returns the Optical Transport Unit (OTU) alarm type for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to O AIS.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F:TYPE?
Parameter(s)	None
Response Syntax	<Alarm>
Response(s)	<p>Alarm:</p> <p>The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the Optical Transport Unit (OTU) alarm type.</p> <p>O AIS, 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>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:TYPE?**

LOF2, Loss of Frame (LOF) is selected as Optical Transport Unit (OTU) alarm.

OOF1, Out of Frame (OOF) is selected as Optical Transport Unit (OTU) alarm.

LOM, Loss of Multiframe (LOM) is selected as Optical Transport Unit (OTU) alarm.

OOM, Out of Multiframe (OOM) is selected as Optical Transport Unit (OTU) alarm.

OBAIE, Optical Transport Unit - Backward Incoming Alignment Error (OTU-BAIE) is selected as Optical Transport Unit (OTU) alarm.

OIAE, Optical Transport Unit - Incoming Alignment Error (OTU-AIE) is selected as Optical Transport Unit (OTU) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OTU1:F:TYPE OAIS
* SOUR:DATA:TEL:OTN:ALAR:OTU1:F:TYPE?
Returns OAIS

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:TYPE

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F**

Description This command enables or disables the status of the Optical Transport Unit (OTU) alarm generation for non standard rates OTU1f/2f.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F<wsp><Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F

Parameter(s)	<p>Set:</p> <p>The program data syntax for <Set> is defined as a <Boolean Program Data> element.</p> <p>The <Set> special forms ON and OFF are accepted on input for increased readability.</p> <p>ON corresponds to 1 and OFF corresponds to 0.</p> <p>Enables or disables the Optical Transport Unit (OTU) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OTU1:F:TYPE OAIS* SOUR:DATA:TEL:OTN:ALAR:OTU1:F ON* SOUR:DATA:TEL:OTN:ALAR:OTU1:F? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:F:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:F?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F?**

Description	This query returns the status of Optical Transport Unit (OTU) alarm generation for non standard rates OTU1f/2f. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F?**

Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of Optical Transport Unit (OTU) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OTU1:F:TYPE OAIS* SOUR:DATA:TEL:OTN:ALAR:OTU1:F ON* SOUR:DATA:TEL:OTN:ALAR:OTU1:F? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:F:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:F

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:MANual:TYPE**

Description This command selects the manual type Optical Transport Unit (OTU) error for non standard rates OTU1f/2f.

At *RST, this value is set to OBIP8.

Syntax :SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:MANual:TYPE<wsp>OBIP8|OBEI|
FAS1|MFAS

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:MANual:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) type error.</p> <p>OBIP8, selects the ODU - Bit Interleave Parity-8 (OTU-BIP-8) as error.</p> <p>OBEI, selects the ODU - Backward Error Indication (OTU-BEI) as error.</p> <p>FAS1, selects the Frame Alignment Signal (FAS) as error.</p> <p>MFAS, selects the Multiframe Alignment Signal (MFAS) as error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE? Returns OBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:MANual:TYPE?</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:MANual:TYPE?**

Description	This query returns the manual type Optical Transport Unit (OTU) error for non standard rates OTU1f/2f. At *RST, this value is set to OBIP8.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:MANual:TYPE?
Parameter(s)	None
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:MANual:TYPE?

Response(s)

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Optical Transport Unit (OTU) type error.

OBIP8, ODU - Bit Interleave Parity-8 (OUT-BIP8) is selected as Optical Transport Unit (OTU) error.

OBEI, ODU - Backward Error Indication (OBE) is selected as Optical Transport Unit (OTU) error.

FAS1, Frame Alignment Signal (FAS) is selected as Optical Transport Unit (OTU) error.

MFAS, Multiframe Alignment Signal (MFAS) is selected as Optical Transport Unit (OTU) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE?
Returns OBIP8

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:MANual:TYPE

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:AMOUNT****Description**

This command sets the amount of Optical Transport Unit (OTU) error to be injected for non standard rates OTU1f/2f.

At *RST, this value is set to 1.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:AMOUNT <wsp> <Amount>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AMOut**

Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the amount of Optical Transport Unit (OTU) error. Choices are 1 through 50.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AMOut?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:F:AMOUnt?**

Description	<p>This query returns the amount of Optical Transport Unit (OTU) error injected for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:F:AMOUnt?[<wsp>MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<Amount>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AMOut?

Response(s)	<p>Amount:</p> <p>The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the amount of Optical Transport Unit (OTU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AMOut</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:INJECT**

Description	<p>This command injects the Optical Transport Unit (OTU) error type for non standard rates OTU1f/2f.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:INJECT
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AMO 15* SOUR:DATA:TEL:OTN:ERR:OTU1:F:INJ
See Also	<ul style="list-style-type: none">* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:MANual:TYPE* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:AMOUNT

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE**

Description This command selects the Optical Transport Unit (OTU) error type for automated injection.

At *RST, this value is set to OBIP8.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE<wsp>OBIP8|
OBEI|FAS1|MFAS

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:AUTomated:TYPE

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) error type for automated injection.</p> <p>OBIP8, selects the ODU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI, selects the ODU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS1, selects the Frame Alignment Signal (FAS) error.</p> <p>MFAS, selects the Multiframe Alignment Signal (MFAS) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:TYPE? Returns OBIP8</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE**

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE?

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:AUTomated:TYPE?**

Description	This query returns the Optical Transport Unit (OTU) error type for automated injection. At *RST, this value is set to OBIP8.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:AUTomated:TYPE?
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE?**

Response(s)

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Optical Transport Unit (OTU) type error for automated injection.

OBIP8, ODU - Bit Interleave Parity-8 (OUT-BIP8) is selected as Optical Transport Unit (OTU) error.

OBEI, ODU - Backward Error Indication (OBE) is selected as Optical Transport Unit (OTU) error.

FAS1, Frame Alignment Signal (FAS) is selected as Optical Transport Unit (OTU) error.

MFAS, Multiframe Alignment Signal (MFAS) is selected as Optical Transport Unit (OTU) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:TYPE?
Returns OBIP8

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:AUTomated:RATE****Description**

This command sets the injection rate for the selected Optical Transport Unit (OTU) error for non standard rates OTU1f/2f.

At *RST, this value is set to 6.5E-05.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:AUTomated:RATE<wsp> <Rate>
|MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:RATE**

Parameter(s) Rate:

The program data syntax for the parameter is defined as a <numeric_value> element. The allowed <numeric_value> elements for this parameter are: MAXimum | MINimum. MAXimum allows to set the instrument to the greatest supported value. MINimum allows to set the instrument to the smallest supported value. Sets the injection rate for the selected Optical Transport Unit (OTU) error. Choices are 1.0E-09 through 6.5E-05.

Example(s)

- * SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:TYPE OBIP8
- * SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:RATE 1.0E-09
- * SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:RATE? Returns 1.0E-09

See Also

- * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated:TYPE
- * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated:RATE?
- * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Optical Transport Unit (OTU) error for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to 6.5E-05.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:AUTomated:RATE? [<wsp> MAXimum MINimum]</pre>
Parameter(s)	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <p>MAXimum MINimum.</p> <p>MAXimum, retrieves the greatest supported value of the instrument.</p> <p>MINimum, retrieves the smallest supported value of the instrument.</p> <p>This parameter is optional. If no token is specified, the current injected rate will be returned.</p>
Response Syntax	<pre><Rate></pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated:RATE?

Response(s)

Rate:

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the injection rate for the selected Optical Transport Unit (OTU) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:RATE
1.0E-09

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:RATE?
Returns 1.0E-09

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated****Description**

This command enables or disables the selected automated Optical Transport Unit (OTU) error at the rate specified or continuously for non standard rates OTU1f/2f.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated<wsp><Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated**

Parameter(s)

Set:

The program data syntax for <Set> is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability.

ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the automated Optical Transport Unit (OTU) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:RATE
1.0E-09

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated?

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:AUTomated?**

Description	This query returns the status of automated Optical Transport Unit (OTU) error injection for non standard rates OTU1f/2f. At *RST, this value is set to OFF.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:AUTomated?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated?**

Response(s)

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of automated Optical Transport Unit (OTU) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:RATE
1.0E-09

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT?

Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:AUTomated:CONTInuous****Description**

This command enables or disables the automated Optical Transport Unit (OTU) error rate injection continuously.

At *RST, this value is set to OFF.

Syntax

```
:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:  
OTU[1..n]:F:AUTomated:CONTInuous <wsp>  
<Set>
```

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AUTomated:CONTInuous

Parameter(s)

Set:

The program data syntax for <Set> is defined as a <Boolean Program Data> element.

The <Set> special forms ON and OFF are accepted on input for increased readability.

ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the automated Optical Transport Unit (OTU) error rate injection continuously.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:CONT
ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:CONT?
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:CONTInuous?

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:AUTomated:CONTInuous?**

Description	This query returns the status of the automated Optical Transport Unit (OTU) error rate injection continuously. At *RST, this value is set to OFF.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:AUTomated:CONTInuous?
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:CONTInuous?**

Response(s)

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of the automated Optical Transport Unit (OTU) error rate injection continuously.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:CONT
ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AUT:CONT?
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:AUTomated:CONTInuous

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:FEC:AUTomated:TYPE****Description**

This command selects the Forward Error Correction (FEC) error type for automated injection.

At *RST, this value is set to FCCW.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:FEC:AUTomated:TYPE<wsp>FCCW
|FUCW|FCSYmb|FCBit|FSCW

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:FEC:AUTomated:TYPE**

Parameter(s)	Error:
	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit FSCWSelects the Forward Error Correction (FEC) error type for automated injection.</p> <p>FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW, selects 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, selects the Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:TYPE**

FSCW, selects the Forward Error Correction - Stress - Codeword (FEC-STRESS-CW) which generates correctable errors composed of a random number of symbol errors (less or equal to 8) containing a random number of bits distributed all over the OTU frame.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:TYPE? Returns FCCW

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AUTomated:TYPE?

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:TYPE?**

Description	This query returns the Forward Error Correction (FEC) error type for automated injection. At *RST, this value is set to FCCW.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:TYPE? <wsp>
Parameter(s)	None
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:FEC:AUTomated:TYPE?

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> 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> <p>FSCW, Forward Error Correction - Stress - Codeword (FEC-STRESS-CW) is selected.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:TYPE? Returns FCCW</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERROR:OTU[1..n]:F:FEC:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERROR:OTU[1..n]:F:FEC:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERROR:OTU[1..n]:F:FEC:AUTomated</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:RATE

Description This command sets the injection rate for the selected Forward Error Correction (FEC) error.

At *RST, this value is set to 1.5E-02.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:RATE<wsp>
MAXimum|MINimum

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:RATE

Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum</p> <p>Sets the injection rate for the selected Forward Error Correction (FEC) error.</p> <p>Choices are 1.0E-07 through 1.5E-02.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:RATE? Returns 1.0E-09</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:E:FEC:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AUTomated:RATE?</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AUTomated</p>

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:FEC:AUTomated:RATE?

Description This query returns the injection rate for the selected Forward Error Correction (FEC) error.

At *RST, this value is set to 1.5E-02.

Syntax :SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:FEC:AUTomated:RATE? <wsp>
MAXimum|MINimum

Parameter(s) None:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum|MINimum
This parameter is optional. If no token is specified, the current injected rate will be returned.

Response Syntax <Rate>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:RATE?****Response(s)**

Rate:

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the injection rate for the selected Forward Error Correction (FEC) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:
RATE 1.0E-09

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:
RATE? Returns 1.0E-09

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated

Description

This command enables or disables the selected automated Forward Error Correction (FEC) error at the rate specified or continuously.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated <wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Enables or disables the automated Forward Error Correction (FEC) error injection.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT: TYPE FCCW * SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT: RATE 1.0E-09 * SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT ON * SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated?**

Description	This query returns the status of the automated Forward Error Correction (FEC) error injection. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated?<wsp>
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of the automated Forward Error Correction (FEC) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:
RATE 1.0E-09

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT
ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT?
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:CONTInuous

Description

This query returns the status of the automated Forward Error Correction (FEC) error injection.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:CONTInuous
<wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:CONTInuous

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Enables or disables the automated Forward Error Correction (FEC) error rate injection continuously.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:CONT ON</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:CONT? Returns 1</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT ON</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AUTomated</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AUTomated:CONTInuous?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AUTomated:CONTInuous?

Description This query returns the status of the automated Forward Error Correction (FEC) error rate injection.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:CONTInuous?
<wsp>

Parameter(s) None

Response Syntax <Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:CONTInuous?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of the automated Forward Error Correction (FEC) error rate injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:
CONT ON

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT:
CONT? Returns 1

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AUT
ON

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AUTomated:CONTInuous

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:TYPE**

Description

This command selects the Optical Payload Unit (OPU) alarm type for non standard rates OTU1f/2f.

At *RST, this value is set to OMSim.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:TYPE<wsp>OMSim

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:TYPE**

Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter is: OMSim.</p> <p>Selects the Optical Payload Unit (OPU) alarm type.</p> <p>OMSim, selects the Optical Payload Unit-Multiplex Structure Identifier Mismatch (OPU-MSIM).</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:OPU1:F:TYPE OMS</p> <p>* SOUR:DATA:TEL:OTN:ALAR:OPU1:F:TYPE?</p> <p>Returns OMSIM</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:TYPE?**

Description	This query returns the Optical Payload Unit (OPU) alarm type for non standard rates OTU1f/2f. At *RST, this value is set to OMSim.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F:TYPE? <wsp>
Parameter(s)	None
Response Syntax	<Alarm>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:TYPE?****Response(s)**

Alarm:

The response data syntax for <Rate> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Optical Payload Unit (OPU) alarm type.

OMSIM, OMSIM is selected as Optical Payload Unit (OPU) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OPU1:F:TYPE OMS

* SOUR:DATA:TEL:OTN:ALAR:OPU1:F:TYPE?

Returns OMSIM

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:TYPE

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F

Description	<p>This command enables or disables the Optical Payload Unit (OPU) alarm generation for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Enables or disables the Optical Payload Unit (OPU) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:F:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1:F ON* SOUR:DATA:TEL:OTN:ALAR:OPU1:F? Returns 1
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:F:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:F?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F**

Description	This query returns the status of the Optical Payload Unit (OPU) alarm generation for non standard rates OTU1f/2f. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F<wsp>
Parameter(s)	None
Response Syntax	<Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F

Response(s)	<p>Set:</p> <p>The response data syntax for <Rate> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the status of the Optical Payload Unit (OPU) alarm generation.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:F:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1:F ON* SOUR:DATA:TEL:OTN:ALAR:OPU1:F? Returns 1
See Also	<p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU [1..n]:F:TYPE</p> <p>SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU [1..n]:F</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TYPE****Description**

This command selects the Optical Data Unit (ODU) alarm type for non standard rates OTU1f/2f.

At *RST, this value is set to OAIS.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TYPE<wsp>OAIS|OBDI|OLCK|
OOCl|OFSF|OBSF|OFSD|OBSD|LOFLom

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARM:
ODU[1..n]:F:TYPE**

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

OAIS|OBDI|OLCK|OOCI|OFSF|OBSF|OFSD|
OBSD|LOFLom

Selects the ODU (Optical Data Unit) alarm type.

OAIS, selects 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.

OBDI, selects 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).

OLCK, selects 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).

OOCI, selects ODU - Backward Defect Indication (ODU-BDI) which generates a "1" in the BDI (byte 3, bit 5) of the PM overhead field continuously.

OFSF, selects ODU - Forward Signal Fail (ODU-FSF) which generates a "00000001" pattern in the FTFL Byte 0 continuously.

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TYPE

OBSF, selects ODU - Backward Signal Fail (ODU-BSF) which generates a "00000001" pattern in the FTFL Byte 128 continuously.

OFSF, selects ODU - Forward Signal Degrade (ODU-FSD) which generates a "00000010" pattern in the FTFL Byte 0 continuously.

OBSD, selects ODU - Backward Signal Degrade (ODU-BSD) which generates a "00000010" pattern in the FTFL Byte 128 continuously.

LOFLom, selects ODU-Loss of Frame Loss of Multiframe (ODU-LOFLOM) which generate error continuously in FAS and MFAS of a multiplexed test case.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TYPE
OFSF

* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TYPE?
Returns OFSF

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TYPE?

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TYPE?**

Description	This query returns the Optical Data Unit (ODU) alarm type for non standard rates OTU1f/2f. At *RST, this value is set to OAIS.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TYPE? <wsp>
Parameter(s)	None
Response Syntax	<Alarm>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARM:
ODU[1..n]:F:TYPE?****Response(s)**

Alarm:

The response data syntax for <Rate> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the Optical Data Unit (ODU) alarm type.

OAIS, ODU - Alarm Indication Signal (ODU-AIS) is selected as Optical Data Unit (ODU) alarm.

OBDI, ODU - Backward Defect Indication (ODU-BDI) is selected as Optical Data Unit (ODU) alarm.

OLCK, ODU - Locked (ODU-LCK) is selected as Optical Data Unit (ODU) alarm.

OOCI, ODU - Open Connection Indication (ODU-OCI) is selected as Optical Data Unit (ODU) alarm.

OFSF, ODU - Forward Signal Fail (ODU-FSF) is selected as Optical Data Unit (ODU) alarm.

OBSF, ODU - Backward Signal Fail (ODU-BSF) is selected as Optical Data Unit (ODU) alarm.

OFSF, ODU - Forward Signal Fail (ODU-FSF) is selected as Optical Data Unit (ODU) alarm.

OBSD, ODU - Backward Signal Degrade (ODU-BSD) is selected as Optical Data Unit (ODU) alarm.

LOFLOM, ODU Loss of Frame Loss of Multiframe (ODU-LOFLOM) is selected as Optical Data Unit (ODU) alarm.

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TYPE?

Example(s) * SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TYPE
 OFSF
 * SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TYPE?
 Returns OFSF

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
 ODU[1..n]:F:TYPE

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F

Description	<p>This command enables or disables the Optical Channel Data Unit (ODU) alarm generation for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F<wsp>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Enables or disables the Optical Data Unit (ODU) alarm generation.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TYPE OFSF</p> <p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:F ON</p> <p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:F? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F

Description	This query returns the status of the Optical Data Unit (ODU) alarm generation for non standard rates OTU1f/2f. At *RST, this value is set to OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F<wsp>
Parameter(s)	None
Response Syntax	<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of Optical Data Unit (ODU) alarm generation.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TYPE
OFSF

* SOUR:DATA:TEL:OTN:ALAR:ODU1:F ON

* SOUR:DATA:TEL:OTN:ALAR:ODU1:F? Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:MANual:TYPE

Description

This command selects the manual type Optical Data Unit (ODU) error for non standard rates OTU1f/2f.

At *RST, this value is set to OBIP8.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:MANual:TYPE<wsp>OBIP8|OBEI

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:MANual:TYPE

Parameter(s)

Error:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8|OBEI

Selects the manual type Optical Data Unit (ODU) error.

OBIP8, selects 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).

OBEI, selects 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.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE?
Returns OBIP8

See Also

*
SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:MANual:TYPE?

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:MANual:TYPE?**

Description	This query returns the manual type Optical Data Unit (ODU) error for non standard rates OTU1f/2f. At *RST, this value is set to OBIP8.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:MANual:TYPE? <wsp>
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:MANual:TYPE?**

Response(s)	<p>Error:</p> <p>The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE? Returns OBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:F:MANual:TYPE</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AMOUnt

Description

This command sets the amount of Optical Data Unit (ODU) error to be injected for non standard rates OTU1f/2f.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AMOUnt<wsp>
MAXimum|MINimum

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AMOut**

Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum</p> <p>Sets the amount of Optical Data Unit (ODU) error. Choices are 1 through 50.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AMO?</p> <p>Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:AMOut?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AMOUnt?

Description	<p>This query returns the amount of Optical Channel Data Unit (ODU) error injected for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AMOUnt? <wsp> MAXimum MINimum</p>
Parameter(s)	<p>None</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum</p> <p>This parameter is optional. If no token is specified, the current amount of error will be returned.</p>
Response Syntax	<p><Amount></p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AMOut?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of Optical Data Unit (ODU) error.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:AMO 15 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:AMO? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AMOut

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:INJect

Description	<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>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:INJect<wsp></pre>
Parameter(s)	None
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AMO 15* SOUR:DATA:TEL:OTN:ERR:ODU1:F:INJ
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AMOUNT

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:FEC:MANual:TYPE****Description**

This command selects the manual type Forward Error Correction (FEC) error for non standard rates OTU1f/2f.

At *RST, this value is set to FCCW.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:FEC:MANual:TYPE<wsp>FCCW|
FUCW|FCSYmb|FCBit|FSCW

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROR:
OTU[1..n]:F:FEC:MANual:TYPE**

Parameter(s)	Error:
	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit FSCW</p> <p>Selects the Forward Error Correction (FEC) error type.</p> <p>FCCW, selects Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW, selects 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, selects Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit, selects Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.</p> <p>FSCW, selects Forward Error Correction - Stress - Codeword (FEC-STRESS-CW) which generates correctable errors composed of a random number of symbol errors (less or equal to 8) containing a random number of bits distributed all over the OTU frame.</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:MANual:TYPE****Example(s)**

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:
TYPE? Returns FCCW

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:MANual:TYPE?

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:FEC:MANual:TYPE?

Description	This query returns the manual type Forward Error Correction (FEC) error for non standard rates OTU1f/2f. At *RST, this value is set to FCCW.
Syntax	:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:FEC:MANual:TYPE? <wsp>
Parameter(s)	None
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:MANual:TYPE?

Response(s)	<p>Error:</p> <p>The response data syntax for <Set> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the Forward Error Correction (FEC) error type.</p> <p>FCCW, Forward Error Correction - Correctable - Codeword (FEC-CORR-CW) is selected as Forward Error Correction (FEC) error.</p> <p>FUCW, Forward Error Correction - Uncorrectable - Codeword (FEC-UNCORR-CW) is selected as Forward Error Correction (FEC) error.</p> <p>FCSYMB, Forward Error Correction - Correctable - Symbol (FEC-CORR-SYMB) is selected as Forward Error Correction (FEC) error.</p> <p>FCBIT, Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) is selected as Forward Error Correction (FEC) error.</p> <p>FSCW, Forward Error Correction - Stress - Codeword (FEC-STRESS-CW) is selected as Forward Error Correction (FEC) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:TYPE? Returns FCCW</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:MANual:TYPE</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AMOut**

Description

This command sets the amount of Forward Error Correction (FEC) error to be injected for non standard rates OTU1f/2f.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AMOut <wsp>
MAXimum | MINimum

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AMOut

Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum</p> <p>Sets the amount of Forward Error Correction (FEC) error.</p> <p>Choices are 1 through 50.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:TYPE FCCW</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AMO?</p> <p>Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:MANual:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AMOut?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:F:FEC:AMOUnt?**

Description This query returns the amount of Forward Error Correction (FEC) error injected for non standard rates OTU1f/2f.

At *RST, this value is set to 1.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:F:FEC:MANual:AMOUnt?<wsp>
MAXimum|MINimum

Parameter(s) None:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum|MINimum
This parameter is optional. If no token is specified, the current amount of error will be returned.

Response Syntax <Amount>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AMOut?****Response(s)**

Amount:

The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the amount of Forward Error Correction (FEC) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AMO 15

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AMO?
Returns 15

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:MANual:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AMOut

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:INJect:

Description	<p>This command injects the Forward Error Correction (FEC) error type.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:INJect:<wsp></pre>
Parameter(s)	None
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN: TYPE FCCW * SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AMO 15 * SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:INJ</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:AMOUNT</pre>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:HISTory?****Description**

This query returns the history status of Optical Transport Unit (OTU) alarm for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:HISTory? <wsp> LOF2 | OOF1 | LOM |
OOM | OAIS | OTIM | OBDI | OIAE | OBlae

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:HISTory?**

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

LOF2|OOF1|LOM|OOM|OAIS|OTIM|OBDI|OIAE|OBlae.

Selects the Optical Transport Unit (OTU) alarm type.

LOF2, selects Loss Of Frame (LOF) when OOF is present for at least 3 ms.

OOF1, selects Out-Of-Frame (OOF) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU (Optical Transport Unit) frames.

LOM, selects Loss Of Multiframe (LOM) when OOM (Out of Multiframe) is present for at least 3 ms.

OOM, selects Out-Of-Multiframe (OOM) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.

OAIS, selects 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.

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:HISTory?**

OTIM, selects 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.

OBDI, selects 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.

OIAE, selects 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.

OBlae, selects 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.

Response Syntax <History>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Optical Transport Unit (OTU) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OTU1:F:TYPE OAIS* SOUR:DATA:TEL:OTN:ALAR:OTU1:F ON* FETC:DATA:TEL:OTN:ALAR:OTU1:F:HIST? OAIS <p>Returns the alarm history.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:F:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]F:

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:SEConds?****Description**

This query returns the number of seconds within which Optical Transport Unit (OTU) alarm occurred for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:SEConds? <wsp>LOF2|OOF1|LOM
|OOM|OAIS|OTIM|OBDI|OIAE|OBlæ

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:SEConds?**

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

LOF2|OOF1|LOM|OOM|OAIS|OTIM|OBDI|OIAE|OBlae

Selects the Optical Transport Unit (OTU) alarm type.

LOF2, selects Loss Of Frame (LOF) when OOF is present for at least 3 ms.

OOF1, selects Out-Of-Frame (OOF) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU (Optical Transport Unit) frames.

LOM, selects Loss Of Multiframe (LOM) when OOM (Out of Multiframe) is present for at least 3 ms.

OOM, selects Out-Of-Multiframe (OOM) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.

**:FETCH[1..n]:DATA:TELEcom:OTN:ALARM:
OTU[1..n]:F:SEConds?**

OAIS, selects 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.

OTIM, selects 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.

OBDI, selects 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.

OIAE, selects 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.

OBlae, selects 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.

Response Syntax <Seconds>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OTU[1..n]:F:SEConds?

Responses

Seconds:

The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the number of seconds of Optical Transport Unit (OTU) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:OTU1:F:TYPE OAIS
* SOUR:DATA:TEL:OTN:ALAR:OTU1:F ON
* FETC:DATA:TEL:OTN:ALAR:OTU1:F:SEC? OAIS
Returns the number of alarmed seconds.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:CURRent?****Description**

This query returns the current status of Optical Transport Unit (OTU) alarm for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:CURRent? <wsp>LOF2|OOF1|LOM
|OOM|O AIS|OTIM|OBDI|OIAE|OBlae

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:CURRENT?**

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

LOF2|OOF1|LOM|OOM|OAIS|OTIM|OBDI|OIAE|OBlae

Selects the Optical Transport Unit (OTU) alarm type.

LOF2, selects Loss Of Frame (LOF) when OOF is present for at least 3 ms.

OOF1, selects Out-Of-Frame (OOF) when FAS (bytes 3, 4, and 5) are in error for at least 5 consecutive OTU (Optical Transport Unit) frames.

LOM, selects Loss Of Multiframe (LOM) when OOM (Out of Multiframe) is present for at least 3 ms.

OOM, selects Out-Of-Multiframe (OOM) when MFAS (Multiframe Alignment Signal) are in error for at least 5 consecutive OTU frames.

**:FETCH[1..n]:DATA:TELEcom:OTN:ALARM:
OTU[1..n]:F:CURRENT?**

OAIS, selects 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.

OTIM, selects 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.

OBDI, selects 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.

OIAE, selects 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.

OBlae, selects 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.

Response Syntax <Current>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OTU[1..n]:F:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of Optical Transport Unit (OTU) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OTU1:F:TYPE OAIS* SOUR:DATA:TEL:OTN:ALAR:OTU1:FON* FETC:DATA:TEL:OTN:ALAR:OTU1:F:CURR? <p>OAIS Returns the current alarm status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:F:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OTU[1..n]:F

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:HISTory?

Description	<p>This query returns the history status of Optical Transport Unit (OTU) error for non standard rates OTU1f/2f.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:HISTory?<wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) error. OBIP8, selects ODU - Bit Interleave Parity-8 (OTU-BIP-8) error. OBEI, selects ODU - Backward Error Indication (OTU-BEI) error. FAS1, selects Frame Alignment Signal (FAS) error. MFAS, selects Multiframe Alignment Signal (MFAS) error.</p>
Response Syntax	<History>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERror:
OTU[1..n]:F:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AMO 15* SOUR:DATA:TEL:OTN:ERR:OTU1:F:INJ* FETC:DATA:TEL:OTN:ERR:OTU1:F:HIST? OBIP8 <p>Returns the error history.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERror:OTU[1..n]:F:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERror:OTU[1..n]:F:AMOUNT* SOURce[1..n]:DATA:TELEcom:OTN:ERror:OTU[1..n]:F:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:F:SEConds?**

Description	<p>This query returns the number of seconds within which Optical Transport Unit (OTU) error occurred for non standard rates OTU1f/2f.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:F:SEConds? <wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) error. OBIP8, selects ODU - Bit Interleave Parity-8 (OTU-BIP-8) error. OBEI, selects ODU - Backward Error Indication (OTU-BEI) error. FAS1, selects Frame Alignment Signal (FAS) error. MFAS, selects Multiframe Alignment Signal (MFAS) error.</p>
Response Syntax	<Seconds>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:SEConds?

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Transport Unit (OTU) error.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:OTU1:F:AMO 15 * SOUR:DATA:TEL:OTN:ERR:OTU1:F:INJ * FETC:DATA:TEL:OTN:ERR:OTU1:F:SEC? OBIP8 Returns the number of errored seconds.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AMOUNT * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:INJect

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:F:CURREnt?

Description	<p>This query returns the current status of Optical Transport Unit (OTU) error for non standard rates OTU1f/2f.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:F:CURREnt? <wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS</p> <p>Selects the Optical Transport Unit (OTU) error. OBIP8, selects ODU - Bit Interleave Parity-8 (OTU-BIP-8) error. OBEI, selects ODU - Backward Error Indication (OTU-BEI) error. FAS1, selects Frame Alignment Signal (FAS) error. MFAS, selects Multiframe Alignment Signal (MFAS) error.</p>
Response Syntax	<Current>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:F:CURREnt?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AMO 15* SOUR:DATA:TEL:OTN:ERR:OTU1:F:INJ* FETC:DATA:TEL:OTN:ERR:OTU1:F:CURR? OBIP8 Returns the current error status.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:OTU[1..n]:F:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:OTU[1..n]:F:AMOUNT* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:OTU[1..n]:F:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:COUNT?**

Description	<p>This query returns the count of Optical Transport Unit (OTU) error for non standard rates OTU1f/2f.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:COUNT?<wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS</p> <p>Selects the Optical Transport Unit (OTU) error.</p> <p>OBIP8, selects ODU - Bit Interleave Parity-8 (OTU-BIP-8) error.</p> <p>OBEI, selects ODU - Backward Error Indication (OTU-BEI) error.</p> <p>FAS1, selects Frame Alignment Signal (FAS) error.</p> <p>MFAS, selects Multiframe Alignment Signal (MFAS) error.</p>
Response Syntax	<pre><Count></pre>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:COUNT?

Response(s)	<p>Count:</p> <p>The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.</p> <p>Returns the count of Optical Transport Unit (OTU) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:OTU1:F:AMO 15* SOUR:DATA:TEL:OTN:ERR:OTU1:F:INJ* FETC:DATA:TEL:OTN:ERR:OTU1:F:COUN? OBIP8 Returns the error count.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:AMOut* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:RATE?**

Description	<p>This query returns the current rate of Optical Transport Unit (OTU) error for non standard rates OTU1f/2f.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:RATE?<wsp>OBIP8 OBEI FAS1 MFAS</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI FAS1 MFAS.</p> <p>Selects the Optical Transport Unit (OTU) error. OBIP8, selects ODU - Bit Interleave Parity-8 (OTU-BIP-8) error. OBEI, selects ODU - Backward Error Indication (OTU-BEI) error. FAS1, selects Frame Alignment Signal (FAS) error. MFAS, selects Multiframe Alignment Signal (MFAS) error.</p>
Response Syntax	<Rate>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:RATE?**

Response(s)	<p>Rate: The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the current rate of Optical Transport Unit (OTU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:OTU1:F:MAN:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:OTU1:F:AMO 15 * SOUR:DATA:TEL:OTN:ERR:OTU1:F:INJ * FETC:DATA:TEL:OTN:ERR:OTU1:F:RATE? OBIP8 Returns the current error rate.</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AMOut * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:INJect</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:HISTory?****Description**

This query returns the history status of Optical Data Unit (ODU) alarm for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:HISTory?<wsp>OAIS|OBDI|OLCK|
OOCl|OFSF|OBSF|OTIM|OFSD|OBSD|LOFLom

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:HISTory?**

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

OAIS|OBDI|OLCK|OOCI|OFSF|OBSF|OTIM|OFSD|OBSD|LOFLom

Selects the Optical Data Unit (ODU) alarm type.

OAIS, selects 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.

OBDI, selects 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.

OLCK, selects ODU - Backward Defect indication (OLCK) indicates that the STAT information detected is "101" for at least 3 consecutive frames.

OOCI, selects ODU - Open Connection Indication (OOCI) indicates that the STAT information detected is "110" for at least 3 consecutive frames.

OFSF, selects ODU - Forward Signal Fail (ODU-FSF) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000001".

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:HISTory?**

OBSF, selects ODU - Backward Signal Fail (ODU-BSF) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000001".

OTIM, selects 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.

OFSF, selects ODU - Forward Signal Degrade (ODU-FSD) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000010"

OBSD, selects ODU - Backward Signal Degrade (ODU-BSD) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000010".

LOFLom, selects ODU-Loss of Frame Loss of Multiframe (ODU-LOFLOM) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.

Response Syntax <History>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TYPE OAIS* SOUR:DATA:TEL:OTN:ALAR:ODU1:E ON* FETC:DATA:TEL:OTN:ALAR:ODU1:F:HIST? OAIS <p>Returns the alarm history.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:F:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:E

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:SEConds?****Description**

This query returns the number of seconds within which Optical Data Unit (ODU) alarm occurred for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

```
:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:  
ODU[1..n]:F:SEConds? <wsp>OAISt|OBDI|OLCK  
|OOCl|OFSF|OBSF|OTIM|OFSD|OBSD|  
LOFLom
```

:FETCh[1..n]:DATA:TELeom:OTN:ALARm: ODU[1..n]:F:SECOndS?

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

OAIS|OBDI|OLCK|OOCI|OFSF|OBSF|OTIM|OFSD|OBSD|LOFLom

Selects the Optical Data Unit (ODU) alarm type.

OAIS, selects 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.

OBDI, selects 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.

OLCK, selects ODU - Backward Defect indication (OLCK) indicates that the STAT information detected is "101" for at least 3 consecutive frames.

OOCI, selects ODU - Open Connection Indication (OOCI) indicates that the STAT information detected is "110" for at least 3 consecutive frames.

OFSF, selects ODU - Forward Signal Fail (ODU-FSF) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000001".

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:SEConds?**

OBSF, selects ODU - Backward Signal Fail (ODU-BSF) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000001".

OTIM, selects 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.

OFSF, selects ODU - Forward Signal Degrade (ODU-FSD) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000010"

OBSD, selects ODU - Backward Signal Degrade (ODU-BSD) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000010".

LOFLom, selects ODU-Loss of Frame Loss of Multiframe (ODU-LOFLOM) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELeom:OTN:ALARm:
ODU[1..n]:F:SEConds?**

Responses(s)

Seconds:

The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the number of seconds of Optical Data Unit (ODU) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TYPE OAIS
* SOUR:DATA:TEL:OTN:ALAR:ODU1:F ON
* FETC:DATA:TEL:OTN:ALAR:ODU1:F:SEC? OAIS
Returns the number of alarmed seconds.

See Also

* SOURce[1..n]:DATA:TELeom:OTN:ALARm:
ODU[1..n]:F:TYPE
* SOURce[1..n]:DATA:TELeom:OTN:ALARm:
ODU[1..n]:F

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:CURRent?****Description**

This query returns the current status of Optical Data Unit (ODU) alarm for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:CURRent? <wsp> OAIS|OBDI|OLCK
|OOCI|OFSF|OBSF|OTIM|OFSD|OBSD|
LOFLom

**:FETCH[1..n]:DATA:TELEcom:OTN:ALARM:
ODU[1..n]:F:CURRENT?**

Parameter(s)

Alarm:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

OAIS|OBDI|OLCK|OOCI|OFSF|OBSF|OTIM|OFSD|OBSD|LOFLom

Selects the Optical Data Unit (ODU) alarm type.

OAIS, selects 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.

OBDI, selects 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.

OLCK, selects ODU - Backward Defect indication (OLCK) indicates that the STAT information detected is "101" for at least 3 consecutive frames.

OOCI, selects ODU - Open Connection Indication (OOCI) indicates that the STAT information detected is "110" for at least 3 consecutive frames.

OFSF, selects ODU - Forward Signal Fail (ODU-FSF) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000001".

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:CURRent?**

OBSF, selects ODU - Backward Signal Fail (ODU-BSF) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000001".

OTIM, selects 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.

OFSD, selects ODU - Forward Signal Degrade (ODU-FSD) is declared when the received FTFL (Fault Type Fault Location) byte 0 is "00000010"

OBSD, selects ODU - Backward Signal Degrade (ODU-BSD) is declared when the received FTFL (Fault Type Fault Location) byte 128 is "00000010".

LOFLom, selects ODU-Loss of Frame Loss of Multiframe (ODU-LOFLOM) which generate error continuously in FAS (Frame Alignment Signal) and MFAS (Multiframe Alignment Signal) of a multiplexed test case.

Response Syntax <Current>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:CURRENT?

Response(s)

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current status of Optical Data Unit (ODU) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TYPE OAIS
* SOUR:DATA:TEL:OTN:ALAR:ODU1:F ON
* FETC:DATA:TEL:OTN:ALAR:ODU1:F:CURR?
OAIS Returns the current alarmed status.

See Also

* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TYPE OAIS
* SOUR:DATA:TEL:OTN:ALAR:ODU1:F ON
* FETC:DATA:TEL:OTN:ALAR:ODU1:F:CURR?
OAIS Returns the current alarmed status.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERror:
ODU[1..n]:F:HISTory?**

Description	<p>This query returns the history status of Optical Data Unit (ODU) error for non standard rates OTU1f/2f.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERror: ODU[1..n]:F:HISTory?<wsp>OBIP8 OBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI</p> <p>Selects the Optical Data Unit (ODU) error type. OBIP8, selects 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, selects 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>
Response Syntax	<History>

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:F:HISTory?

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Optical Data Unit (ODU) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AMO 15* SOUR:DATA:TEL:OTN:ERR:ODU1:F:INJ* FETC:DATA:TEL:OTN:ERR:ODU1:F:HIST? OBIP8 <p>Returns the error history.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:ODU[1..n]:F:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:ODU[1..n]:F:AMOUNT* SOURce[1..n]:DATA:TELEcom:OTN:ERROr:ODU[1..n]:F:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:SECOnds?**

Description	<p>This query returns the number of seconds within which Optical Data Unit (ODU) error occurred for non standard rates OTU1f/2f.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:F:SECOnds? <wsp>OBIP8 OBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8 OBEI</p> <p>Selects the Optical Data Unit (ODU) error type.</p> <p>OBIP8, selects 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, selects 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>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:SEConds?

Response Syntax <Seconds>

Response(s) Seconds:
The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Returns the number of seconds of Optical Data Unit (ODU) error.

Example(s) * SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AMO 15
* SOUR:DATA:TEL:OTN:ERR:ODU1:F:INJ
* FETC:DATA:TEL:OTN:ERR:ODU1:F:SEC? OBIP8
Returns the number of errored seconds.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:CURREnt?**

Description This query returns the current status of Optical Data Unit (ODU) error for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:CURREnt?<wsp>OBIP8|OBEI

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8|OBEI
Selects the Optical Data Unit (ODU) error type. OBIP8, selects 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).
OBEI, selects 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.

Response Syntax <Current>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:CURRent?

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of Optical Data Unit (ODU) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AMO 15* SOUR:DATA:TEL:OTN:ERR:ODU1:F:INJ* FETC:DATA:TEL:OTN:ERR:ODU1:F:CURR? OBIP8 Returns the current error status.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AMOUNT* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:COUNT?**

Description This query returns the count of Optical Data Unit (ODU) error for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:COUNT? <wsp>OBIP8|OBEI

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OBIP8|OBEI
Selects the Optical Data Unit (ODU) error type.
OBIP8, selects 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).
OBEI, selects 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.

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:COUNT?**

Response Syntax <Count>

Response(s) Count:
The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.
Returns the count of Optical Data Unit (ODU) error.

Example(s) * SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE
OBIP8
* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AMO 15
* SOUR:DATA:TEL:OTN:ERR:ODU1:F:INJ
* FETC:DATA:TEL:OTN:ERR:ODU1:F:COUN?
OBIP8 Returns the error count.

See Also * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:MANual:TYPE

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:RATE?**

Description	<p>This query returns the current rate of Optical Data Unit (ODU) error for non standard rates OTU1f/2f.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:F:RATE?<wsp>OBIP8 OBEI</code>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter are: OBIP8 OBEI.</p> <p>Selects the Optical Data Unit (ODU) error type. OBIP8, selects 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, selects 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>
Response Syntax	<code><Rate></code>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of Optical Data Unit (ODU) error.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AMO 15* SOUR:DATA:TEL:OTN:ERR:ODU1:F:INJ* FETC:DATA:TEL:OTN:ERR:ODU1:F:RATE? OBIP8 Returns the current error rate.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AMOUNT* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:INJect

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:F:AUTomated:TYPE****Description**

This command selects the Optical Data Unit (ODU) error type for automated injection.

At *RST, this value is set to OBIP8.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:ODU
[1..n]:F:AUTomated:TYPE<wsp>OBIP8|OBEI

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TYPE

Parameter(s)	<p>Error:</p> <p>The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> element for this parameter are: OBIP8 OBEI.</p> <p>Selects the Optical Data Unit (ODU) error type. OBIP8, selects 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, selects 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TYPE OBIP8* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TYPE? Returns OBIP8
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:AUTomated:TYPE?* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:AUTomated:RATE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:AUTomated

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:TYPE?**

Description	This query returns the Optical Data Unit (ODU) error type for automated injection. At *RST, this value is set to OBIP8.
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU [1..n]:F:AUTomated:TYPE? <wsp>
Parameter(s)	None
Response Syntax	<Error>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TYPE?

Response(s)

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the Optical Data Unit (ODU) error type for automated injection.

OBIP8, ODU - Bit Interleave Parity-8 (ODU-BIP8) is selected as Optical Data Unit (ODU) error.

OBEI, ODU - Backward Error Indication (ODU-BEI) is selected as Optical Data Unit (ODU) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TYPE
OBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TYPE?
Returns OBIP8

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:RATE****Description**

This command sets the injection rate for the selected Optical Data Unit (ODU) error.

At *RST, this value is device dependent.

Syntax

SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:RATE <wsp> MAXimum
| MINimum

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:RATE**

Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Sets the injection rate for the selected Optical Data Unit (ODU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:RATE? Returns 1.0E-09</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:RATE?</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated</p>

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:RATE?**

Description	<p>This query returns the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:RATE? <wsp>MAXimu m MINimum</p>
Parameter(s)	<p>None:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>This parameter is optional. If no token is specified, the injected rate will be returned.</p>
Response Syntax	<p><Rate></p>

SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:RATE?

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Optical Data Unit (ODU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:RATE? Returns 1.0E-09</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated</p>

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated**

Description	<p>This command enables or disables the selected automated Optical Data Unit (ODU) error at the rate specified or continuously for non standard rates OTU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated<wsp><Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:RATE 1.0E-09 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT ON * SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:RATE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated?</pre>

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated?**

Description	This query returns the status of the automated Optical Data Unit (ODU) error injection for non standard rates OTU1f/2f. At *RST, this value is set to OFF.
Syntax	SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated?<wsp>
Parameter(s)	None
Response Syntax	<Set>

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated?**

Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the automated Optical Data Unit (ODU) error injection.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TYPE OBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT ON</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT? Returns 1</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:RATE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated</p>

SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:CONTInuous

Description

This command enables or disables the automated Optical Data Unit (ODU) error injection rate continuously for non standard rates OTU1f/2f.

At *RST, this value is set to OFF.

Syntax

SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:CONTInuous<wsp>
<Set>

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:CONTInuous**

Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. Enables or disables the automated Optical Data Unit (ODU) error injection rate.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:CONT ON * SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:CONT? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:CONTInuous?

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:CONTInuous?**

Description	This query returns the status of the automated Optical Data Unit (ODU) error injection rate for non standard rates OTU1f/2f. At *RST, this value is set to OFF.
Syntax	SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated?CONTInuous?<wsp>
Parameter(s)	None
Response Syntax	<Set>

**SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:CONTInuous?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the automated Optical Data Unit (ODU) error injection rate.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:CONT ON * SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:CONT? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:CONTInuous

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:HISTory?**

Description This query returns the history status of Optical Payload Unit (OPU) alarm for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:HISTory? <wsp>OPLM|OMSim

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPLM|OMSim
Selects the Optical Payload Unit (OPU) alarm type.
OPLM, selects the Optical Payload Unit Payload Mismatch (OPU-PLM) alarm.
OMSim, selects the Optical Payload Unit Multiplex Structure Identifier Mismatch (OPU-MSIM) alarm.

Response Syntax <History>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:F:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1:F ON* FETC:DATA:TEL:OTN:ALAR:OPU1:F:HIST? OMS <p>Returns the alarm history.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:F:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:F

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F:SEConds?

Description This query returns the number of seconds within which Optical Payload Unit (OPU) alarm occurred for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:SEConds? <wsp>OPLM|OMSim

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPLM|OMSim
Selects the Optical Payload Unit (OPU) alarm type.
OPLM, selects the Optical Payload Unit Payload Mismatch (OPU-PLM) alarm.
OMSim, selects the Optical Payload Unit Multiplex Structure Identifier Mismatch (OPU-MSIM) alarm.

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Payload Unit (OPU) alarm.
Example(s)	* SOUR:DATA:TEL:OTN:ALAR:OPU1:F:TYPE OMS * SOUR:DATA:TEL:OTN:ALAR:OPU1:F ON * FETC:DATA:TEL:OTN:ALAR:OPU1:F:SEC? OMS Returns the number of alarmed seconds.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: OPU[1..n]:F:CURRent?

Description This query returns the current status of Optical Payload Unit (OPU) alarm for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:CURRent? <wsp>OPLM|OMSim

Parameter(s) Alarm:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: OPLM|OMSim
Selects the Optical Payload Unit (OPU) alarm type.
OPLM, selects the Optical Payload Unit Payload Mismatch (OPU-PLM) alarm.
OMSim, selects the Optical Payload Unit Multiplex Structure Identifier Mismatch (OPU-MSIM) alarm.

Response Syntax <Current>

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
OPU[1..n]:F:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of 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 is not running.</p>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ALAR:OPU1:F:TYPE OMS* SOUR:DATA:TEL:OTN:ALAR:OPU1:F ON* FETC:DATA:TEL:OTN:ALAR:OPU1:F:CURR? <p>OMS Returns the current alarm status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:F:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:OPU[1..n]:F

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:HISTory?**

Description

This query returns the history status of Forward Error Correction (FEC) error for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:HISTory?<wsp>FCCW|FUCW|
FCSYmb|FCBit

:FETCh[1..n]:DATA:TELecom:OTN:ERROr: OTU[1..n]:F:FEC:HISTOrY?

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit.</p> <p>Selects Forward Error Correction (FEC) error type.</p> <p>FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW, selects 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, selects the Forward Error Correction - Correctable - Symbol (FEC-UNCORR-CW) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.</p>
Response Syntax	<History>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:HISTory?

Response(s)

History:

The response data syntax for <History> 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.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AMO 15

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:INJ

* FETC:DATA:TEL:OTN:ERR:OTU1:F:FEC:HIST? FCCW Returns the error history.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:MANual:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AMOUNT

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:SEConds?****Description**

This query returns the number of seconds within which Forward Error Correction (FEC) error occurred for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:SEConds?<wsp>FCCW|FUCW
|FCSYmb|FCBit

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
OTU[1..n]:F:FEC:SECOnds?**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit</p> <p>Selects Forward Error Correction (FEC) error type.</p> <p>FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW, selects 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, selects the Forward Error Correction - Correctable - Symbol (FEC-UNCORR-CW) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.</p>
Response Syntax	<Seconds>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:SEConds?**

Response(s)	<p>Seconds:</p> <p>The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current status of Forward Error Correction (FEC) 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>
Example(s)	<ul style="list-style-type: none">* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:TYPE FCCW* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AMO 15* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:INJ* FETC:DATA:TEL:OTN:ERR:OTU1:F:FEC:SEC? FCCW Returns the number of errored seconds.
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AMOUNT* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:CURREnt?**

Description

This query returns the current status of Forward Error Correction (FEC) error for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:CURREnt?<wsp>FCCW|FUCW
|FCSYmb|FCBit

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: OTU[1..n]:F:FEC:CURREnt?

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit.</p> <p>Selects Forward Error Correction (FEC) error type.</p> <p>FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW, selects 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, selects the Forward Error Correction - Correctable - Symbol (FEC-UNCORR-CW) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.</p>
Response Syntax	<Current>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:CURREnt?

Responses

Current:

The response data syntax for <Current> 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 is not running.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:
TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AMO 15
* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:F:FEC:CURREnt?
FCCW Returns the current error status.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:INJECT

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:COUNT?****Description**

This query returns the count of Forward Error Correction (FEC) error for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:COUNT? <wsp>FCCW|FUCW|
FCSYmb|FCBit

**:FETCh[1..n]:DATA:TELecom:OTN:ERROr:
OTU[1..n]:F:FEC:COUNT?**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit</p> <p>Selects Forward Error Correction (FEC) error type.</p> <p>FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW, selects 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, selects the Forward Error Correction - Correctable - Symbol (FEC-UNCORR-CW) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.</p>
Response Syntax	<Count>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:COUNT?****Response(s)**

Count:

The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element.

Returns the count of Forward Error Correction (FEC) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:
TYPE FCCW

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AMO 15

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:INJ

* FETC:DATA:TEL:OTN:ERR:OTU1:F:FEC:COUN?
FCCW Returns the error count.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:MANual:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:AMOUNT

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:INJECT

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:RATE?**

Description

This query returns the current rate of Forward Error Correction (FEC) error for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
OTU[1..n]:F:FEC:RATE? <wsp> FCCW|FUCW|FC
SYmb|FCBit

:FETCh[1..n]:DATA:TELEcom:OTN:ERROR: OTU[1..n]:F:FEC:RATE?

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: FCCW FUCW FCSYmb FCBit</p> <p>Selects Forward Error Correction (FEC) error type.</p> <p>FCCW, selects the Forward Error Correction - Correctable - Codeword (FCCORR-CW) which generates 8 symbols (bytes) containing 8 bits in error each, in each codeword.</p> <p>FUCW, selects 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, selects the Forward Error Correction - Correctable - Symbol (FEC-UNCORR-CW) which generates 1 symbol (byte) containing 8 bits in error.</p> <p>FCBit, selects the Forward Error Correction - Correctable - Bit (FEC-CORR-BIT) which generates 1 symbol (byte) containing 1bit in error.</p>
Response Syntax	<Rate>

:FETCh[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:FEC:RATE?

Response(s)

Rate:

The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the current rate of Forward Error Correction (FEC) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:MAN:TYPE FCCW
* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:AMO 15
* SOUR:DATA:TEL:OTN:ERR:OTU1:F:FEC:INJ
* FETC:DATA:TEL:OTN:ERR:OTU1:F:FEC:RATE?
FCCW Returns the current error rate.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:MANual:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:AMOUNT
* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:F:FEC:INJECT

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:F:OVERhead**

Description This command sets the overhead byte values for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:F: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]:F:OVERhead**

Parameter(s)

Overhead:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

OA111|OA112|OA113|OA214|OA215|OA216|MFAS17|SM18|SM19|SM110|GCC0111|GCC0112|RES113|RES114

Selects overhead bytes for non standard rates OTU1f/2f.

OA111, selects OA1 as overhead byte.

OA112, selects OA1 as overhead byte.

OA113, selects OA1 as overhead byte.

OA214, selects OA2 as overhead byte.

OA215, selects OA2 as overhead byte.

OA216, selects OA2 as overhead byte.

MFAS17, selects MFAS as overhead byte.

SM18, selects SM as overhead byte.

SM19, selects SM as overhead byte.

SM110, selects SM as overhead byte.

GCC0111, selects GCC0 as overhead byte.

GCC0112, selects GCC0 as overhead byte.

RES113, selects RES as overhead byte.

RES114, selects RES as overhead byte.

Value:

The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the overhead byte values.

**:SOURce[1..n]:DATA:TELecom:OTN:OH:
OTU[1..n]:F:OVERhead****Example(s)**

* SOUR:DATA:TEL:OTN:OH:OTU1:F:OVER
OA111, #HF6
* SOUR:DATA:TEL:OTN:OH:OTU1:F:OVER?
OA111 Returns #HF6

See Also

* SOURce[1..n]:DATA:TELecom:OTN:OH:
OTU[1..n]:F:OVERhead?

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:F:OVERhead?

Description

This query returns the overhead byte values for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:  
OTU[1..n]:F:OVERhead?<wsp>OA111|OA112|O  
A113|OA214|OA215|OA216|MFAS17|SM18|SM1  
9|SM110|GCC0111|GCC0112|RES113|RES114
```

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:F:OVERhead?

Parameter(s)	<p>Overhead:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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 OTU1f/2f.</p> <p>OA111, selects OA1 as overhead byte. OA112, selects OA1 as overhead byte. OA113, selects OA1 as overhead byte. OA214, selects OA2 as overhead byte. OA215, selects OA2 as overhead byte. OA216, selects OA2 as overhead byte. MFAS17, selects MFAS as overhead byte. SM18, selects SM as overhead byte. SM19, selects SM as overhead byte. SM110, selects SM as overhead byte. GCC0111, selects GCC0 as overhead byte. GCC0112, selects GCC0 as overhead byte. RES113, selects RES as overhead byte. RES114, selects RES as overhead byte.</p>
Response Syntax	<Value>

:SOURce[1..n]:DATA:TELecom:OTN:OH: OTU[1..n]:F:OVERhead?

Response(s)	Value: The response data syntax for <Current> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.Value Returns the overhead byte values in hexadecimal format.
Example(s)	* SOUR:DATA:TEL:OTN:OH:OTU1:F:OVER OA111, #HF6 * SOUR:DATA:TEL:OTN:OH:OTU1:F:OVER? OA111 Returns #HF7
See Also	* SOURce[1..n]:DATA:TELecom:OTN:OH: OTU[1..n]:F:OVERhead

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OTU[1..n]:F:DEFault**

Description	<p>This command resets or overwrites the overhead byte values for non standard rates OTU1f/2f.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:F:DEFault<wsp>
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:OH:OTU1:F:DEF

:SENSe[1..n]:DATA:TELEcom:OTN:OH: OTU[1..n]:F:OVERhead?

Description	<p>This query returns the overhead byte values for non standard rates OTU1f/2f.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:OH:OTU[1..n] :F:OVERhead? <wsp>OA111 OA112 OA113 OA2 14 OA215 OA216 MFAS17 SM18 SM19 SM110 GCC0111 GCC0112 RES113 RES114</pre>
Parameter(s)	<p>Overhead:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:</p> <pre>OA111 OA112 OA113 OA214 OA215 OA216 MF AS17 SM18 SM19 SM110 GCC0111 GCC0112 R ES113 RES114</pre> <p>Selects overhead bytes for non standard rates OTU1f/2f.</p> <p>OA111, selects OA1 as overhead byte. OA112, selects OA1 as overhead byte. OA113, selects OA1 as overhead byte. OA214, selects OA2 as overhead byte. OA215, selects OA2 as overhead byte.</p>
Response Syntax	<Value>

**:SENSe[1..n]:DATA:TELecom:OTN:OH:
OTU[1..n]:F:OVERhead?**

Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.</p> <p>Returns the overhead byte values in hexadecimal format.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:OH:OTU1:F:OVER? OA111</p> <p>Returns the OTU overhead byte values.</p>
See Also	<p>* SOURce[1..n]:DATA:TELecom:OTN:OH: OTU[1..n]:F:OVERhead</p>

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:OVERhead

Description

This command sets the overhead byte values for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:  
ODU[1..n]:F:OVERhead <wsp> RES21 | RES22 | RE  
S23 | TCMACT24 | TCM625 | TCM626 | TCM627 | TCM  
528 | TCM529 | TCM5210 | TCM4211 | TCM4212 | TCM  
4213 | FTFL214 | TCM331 | TCM332 | TCM333 | TCM2  
34 | TCM235 | TCM236 | TCM137 | TCM138 | TCM139 |  
PM310 | PM311 | PM312 | EXP313 | EXP314 | GCC141  
| GCC142 | GCC243 | GCC244 | APSPCC45 |  
APSPCC46 | APSPCC47 | APSPCC48 | RES49 | RES41  
0 | RES411 | RES412 | RES413 | RES414, <Value >
```


:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:OVERhead

Parameter(s)

Overhead:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> 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.

Selects overhead bytes for non standard rates OTU1f/2f.

RES21, selects RES as overhead byte.

RES22, selects RES as overhead byte.

RES23, selects RES as overhead byte.

TCMACT24, selects TCM as overhead byte.

TCM625, selects TCM6 as overhead byte.

TCM626, selects TCM6 as overhead byte.

TCM627, selects TCM6 as overhead byte.

TCM528, selects TCM5 as overhead byte.

TCM529, selects TCM5 as overhead byte.

TCM5210, selects TCM5 as overhead byte.

TCM4211, selects TCM4 as overhead byte.

TCM4212, selects TCM4 as overhead byte.

TCM4213, selects TCM4 as overhead byte.

FTFL214, selects FTFL as overhead byte.

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:OVERhead

TCMACT331, selects TCM3 as overhead byte.
TCM332, selects TCM3 as overhead byte.
TCM333, selects TCM3 as overhead byte.
TCM234, selects TCM2 as overhead byte.
TCM235, selects TCM2 as overhead byte.
TCM236, selects TCM2 as overhead byte.
TCM137, selects TCM1 as overhead byte.
TCM138, selects TCM1 as overhead byte.
TCM139, selects TCM1 as overhead byte.
PM310, selects PM as overhead byte.
PM311, selects PM as overhead byte.
PM312, selects PM as overhead byte.
EXP313, selects EXP as overhead byte.
EXP314, selects EXP as overhead byte.
GCC141, selects GCC1 as overhead byte.
GCC142, selects GCC1 as overhead byte.
GCC243, selects GCC2 as overhead byte.
GCC244, selects GCC2 as overhead byte.
APSPCC45, selects APSPCC as overhead byte.
APSPCC46, selects APSPCC as overhead byte.
APSPCC47, selects APSPCC as overhead byte.
APSPCC48, selects APSPCC as overhead byte.
RES49, selects RES as overhead byte.
RES410, selects RES as overhead byte.
RES411, selects RES as overhead byte.
RES412, selects RES as overhead byte.
RES413, selects RES as overhead byte.
RES414, selects RES as overhead byte.

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:F:OVERhead****Value:**

The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the overhead byte values.

Example(s)

* SENS:DATA:TEL:OTN:OH:OTU1:F:OVER? OA111
Returns the OTU overhead byte values.

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:F:OVERhead?

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:OVERhead?

Description

This query returns the overhead byte values for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:  
ODU[1..n]:F:OVERhead?<wsp>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
```

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:OVERhead?

Parameter(s)

Overhead:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> 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.

Selects overhead bytes for non standard rates OTU1f/2f.

RES21, selects RES as overhead byte.

RES22, selects RES as overhead byte.

RES23, selects RES as overhead byte.

TCMACT24, selects TCM as overhead byte.

TCM625, selects TCM6 as overhead byte.

TCM626, selects TCM6 as overhead byte.

TCM627, selects TCM6 as overhead byte.

TCM528, selects TCM5 as overhead byte.

TCM529, selects TCM5 as overhead byte.

TCM5210, selects TCM5 as overhead byte.

TCM4211, selects TCM4 as overhead byte.

TCM4212, selects TCM4 as overhead byte.

TCM4213, selects TCM4 as overhead byte.

FTFL214, selects FTFL as overhead byte.

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:OVERhead?

GCC2, selects GCC2 as overhead byte.
TCMACT331, selects TCM3 as overhead byte.
TCM332, selects TCM3 as overhead byte.
TCM333, selects TCM3 as overhead byte.
TCM234, selects TCM2 as overhead byte.
TCM235, selects TCM2 as overhead byte.
TCM236, selects TCM2 as overhead byte.
TCM137, selects TCM1 as overhead byte.
TCM138, selects TCM1 as overhead byte.
TCM139, selects TCM1 as overhead byte.
PM310, selects PM as overhead byte.
PM311, selects PM as overhead byte.
PM312, selects PM as overhead byte.
EXP313, selects EXP as overhead byte.
EXP314, selects EXP as overhead byte.
GCC141, selects GCC1 as overhead byte.
GCC142, selects GCC1 as overhead byte.
GCC243, selects GCC2 as overhead byte.
GCC244, selects GCC2 as overhead byte.
APSPCC45, selects APSPCC as overhead byte.
APSPCC46, selects APSPCC as overhead byte.
APSPCC47, selects APSPCC as overhead byte.
APSPCC48, selects APSPCC as overhead byte.
RES49, selects RES as overhead byte.
RES410, selects RES as overhead byte.
RES411, selects RES as overhead byte.
RES412, selects RES as overhead byte.
RES413, selects RES as overhead byte.
RES414, selects RES as overhead byte.

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:F:OVERhead?****Response Syntax** <Value>**Response(s)**

Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the overhead byte values in hexadecimal format.

Example(s)

* SOUR:DATA:TEL:OTN:OH:ODU1:F:OVER
RES21, #HF6

* SOUR:DATA:TEL:OTN:OH:ODU1:F:OVER?
RES21 Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:F:OVERhead

:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:DEFault

Description	<p>This command resets or overwrites the overhead byte values for non standard rates OTU1f/2f.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:DEFault<wsp></pre>
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:OH:ODU1:F:DEF

:SENSe[1..n]:DATA:TELEcom:OTN:OH: ODU[1..n]:F:OVERhead?

Description

This query returns the overhead byte values for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

```
:SENSe[1..n]:DATA:TELEcom:OTN:OH:ODU[1..n]
:F:OVERhead? <wsp>RES21|RES22|RES23|TCM
ACT24|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
```

**:SENSE[1..n]:DATA:TELEcom:OTN:OH:
ODU[1..n]:F:OVERhead?**

Parameter(s)

Overhead:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> 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.

Selects overhead bytes.

RES21, selects RES as overhead byte.

RES22, selects RES as overhead byte.

RES23, selects RES as overhead byte.

TCMACT24, selects TCM as overhead byte.

TCM625, selects TCM6 as overhead byte.

TCM626, selects TCM6 as overhead byte.

TCM627, selects TCM6 as overhead byte.

TCM528, selects TCM5 as overhead byte.

TCM529, selects TCM5 as overhead byte.

TCM5210, selects TCM5 as overhead byte.

**:SENSe[1..n]:DATA:TELeom:OTN:OH:
ODU[1..n]:F:OVERhead?**

TCM4211, selects TCM4 as overhead byte.
TCM4212, selects TCM4 as overhead byte.
TCM4213, selects TCM4 as overhead byte.
FTFL214, selects FTFL as overhead byte.
GCC2, selects GCC2 as overhead byte.
TCM331, selects TCM3 as overhead byte.
TCM332, selects TCM3 as overhead byte.
TCM333, selects TCM3 as overhead byte.
TCM234, selects TCM2 as overhead byte.
TCM235, selects TCM2 as overhead byte.
TCM236, selects TCM2 as overhead byte.
TCM137, selects TCM1 as overhead byte.
TCM138, selects TCM1 as overhead byte.
TCM139, selects TCM1 as overhead byte.
PM310, selects PM as overhead byte.
PM311, selects PM as overhead byte.
PM312, selects PM as overhead byte.
EXP313, selects EXP as overhead byte.
EXP314, selects EXP as overhead byte.
GCC141, selects GCC1 as overhead byte.
GCC142, selects GCC1 as overhead byte.
GCC243, selects GCC2 as overhead byte.
GCC244, selects GCC2 as overhead byte.
APSPCC45, selects APSPCC as overhead byte.

:SENSe[1..n]:DATA:TELecom:OTN:OH: ODU[1..n]:F:OVERhead?

APSPCC46, selects APSPCC as overhead byte.

APSPCC47, selects APSPCC as overhead byte.

APSPCC48, selects APSPCC as overhead byte.

RES49, selects RES as overhead byte.

RES410, selects RES as overhead byte.

RES411, selects RES as overhead byte.

RES412, selects RES as overhead byte.

RES413, selects RES as overhead byte.

RES414, selects RES as overhead byte.

Response Syntax <Value>

Response(s) Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the overhead byte values in hexadecimal format.

Example(s) * SENS:DATA:TEL:OTN:OH:ODU1:F:OVER? RES21
Returns the ODU overhead byte values.

See Also * SOURce[1..n]:DATA:TELecom:OTN:OH:
ODU[1..n]:F:OVERhead

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:F:OVERhead****Description**

This command sets the overhead byte values for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:  
OPU[1..n]:F:OVERhead<wsp>RES115|RES116|J  
C116|RES215|RES216|JC216|RES315|RES316|J  
C316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|PSI5  
|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|PSI13  
|PSI14|PSI15|PSI16|PSI17, <Value>
```

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:F:OVERhead**

Parameter(s)	Overhead:
	<p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: RES115 RES116 JC116 RES215 RES216 JC216 RES315 RES316 JC316 PSI415 NJO416 PSI0 PSI2 PSI3 PSI4 PSI5 PSI6 PSI7 PSI8 PSI9 PSI10 PSI11 PSI12 PSI13 PSI14 PSI15 PSI16 PSI17.</p> <p>Selects overhead bytes for non standard rates OTU1f/2f.</p> <p>RES115, selects RES as overhead byte. RES116, selects RES as overhead byte. JC116, selects JC as overhead byte. RES215, selects RES as overhead byte. RES216, selects RES as overhead byte. JC216, selects JC as overhead byte. RES315, selects RES as overhead byte. RES316, selects RES as overhead byte. JC316, selects JC as overhead byte.</p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:F:OVERhead**

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Value:

The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element.

Sets the overhead byte values.

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:F:OVERhead

Example(s)

* SOUR:DATA:TEL:OTN:OH:OPU1:F:OVER
RES115, #HF6

* SOUR:DATA:TEL:OTN:OH:OPU1:F:OVER?
RES115 Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:F:OVERhead?

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:F:OVERhead?****Description**

This query returns the overhead byte values for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax

```
:SOURce[1..n]:DATA:TELEcom:OTN:OH:OPU[1..n]:F:OVERhead?<wsp>RES115|RES116|JC116|RES215|RES216|JC216|RES315|RES316|JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|PSI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|PSI13|PSI14|PSI15|PSI16|PSI17
```

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:F:OVERhead?**

Parameter(s)

Overhead:

The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

RES115 | RES116 | JC116 | RES215 | RES216 | JC216 |
RES315 | RES316 | JC316 | PSI415 | NJO416 | PSI0 | PS
I2 | PSI3 | PSI4 | PSI5 | PSI6 | PSI7 | PSI8 | PSI9 | PSI10 |
PSI11 | PSI12 | PSI13 | PSI14 | PSI15 | PSI16 | PSI17.

Selects overhead bytes.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

**:SOURce[1..n]:DATA:TELecom:OTN:OH:
OPU[1..n]:F:OVERhead?**

PSI9, selects PSI9 as overhead byte.
PSI10, selects PSI10 as overhead byte.
PSI11, selects PSI11 as overhead byte.
PSI12, selects PSI12 as overhead byte.
PSI13, selects PSI13 as overhead byte.
PSI14, selects PSI14 as overhead byte.
PSI15, selects PSI15 as overhead byte.
PSI16, selects PSI16 as overhead byte.
PSI17, selects PSI17 as overhead byte.

Response Syntax <Value>

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:F:OVERhead?**

Response(s)

Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the overhead byte values in hexadecimal format.

Returns the overhead byte values in hexadecimal format.

Returns the received overhead byte values for non standard rates OPU1f/2f.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

**:SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:F:OVERhead?**

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Example(s)

* SOUR:DATA:TEL:OTN:OH:OPU1:F:OVER
RES115, #HF6

* SOUR:DATA:TEL:OTN:OH:OPU1:F:OVER?
RES115 Returns #HF6

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:OH:
OPU[1..n]:F:OVERhead

:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:F:DEFault

Description	<p>This command resets or overwrites the overhead byte values for non standard rates OTU1f/2f.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:F:DEFault<wsp></pre>
Parameter(s)	None
Example(s)	* SOUR:DATA:TEL:OTN:OH:OPU1:F:DEF

:SENSe[1..n]:DATA:TELecom:OTN:OH: OPU[1..n]:F:OVERhead?

Description This query returns the regenerator overhead byte values for non standard rates OTU1f/2f.

At *RST, this value is device dependent.

Syntax :SENSe[1..n]:DATA:TELecom:OTN:OH:
OPU[1..n]:F:OVERhead?<wsp>RES115|RES116|
JC116|RES215|RES216|JC216|RES315|RES316|
JC316|PSI415|NJO416|PSI0|PSI2|PSI3|PSI4|
PSI5|PSI6|PSI7|PSI8|PSI9|PSI10|PSI11|PSI12|
PSI13|PSI14|PSI15|PSI16|PSI17

**:SENSe[1..n]]:DATA:TELEcom:OTN:OH:
OPU[1..n]:F:OVERhead?**

Parameter(s)

Overhead:

The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:

RES115|RES116|JC116|RES215|RES216|JC216|
RES315|RES316|JC316|PSI415|NJO416|PSI0|
PSI2|PSI3|PSI4|PSI5|PSI6|PSI7|PSI8|PSI9|
PSI10|PSI11|PSI12|PSI13|PSI14|PSI15|PSI16|
PSI17.

Selects overhead bytes.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

**:SENSe[1..n]:DATA:TELecom:OTN:OH:
OPU[1..n]:F:OVERhead?**

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Response Syntax <Value>

:SENSe[1..n]:DATA:TELEcom:OTN:OH: OPU[1..n]:F:OVERhead?

Response(s)

Value:

The response data syntax for <Value> is defined as a <HEXADECIMAL NUMERIC RESPONSE DATA> element.

Returns the overhead byte values in hexadecimal format.

Returns the overhead byte values in hexadecimal format.

Returns the received overhead byte values for non standard rates OPU1f/2f.

RES115, selects RES as overhead byte.

RES116, selects RES as overhead byte.

JC116, selects JC as overhead byte.

RES215, selects RES as overhead byte.

RES216, selects RES as overhead byte.

JC216, selects JC as overhead byte.

RES315, selects RES as overhead byte.

RES316, selects RES as overhead byte.

JC316, selects JC as overhead byte.

PSI415, selects PSI as overhead byte.

NJO416, selects NJO as overhead byte.

PSI0, selects PSI0 as overhead byte.

PSI2, selects PSI2 as overhead byte.

PSI3, selects PSI3 as overhead byte.

PSI4, selects PSI4 as overhead byte.

PSI5, selects PSI5 as overhead byte.

PSI6, selects PSI6 as overhead byte.

PSI7, selects PSI7 as overhead byte.

PSI8, selects PSI8 as overhead byte.

PSI9, selects PSI9 as overhead byte.

PSI10, selects PSI10 as overhead byte.

**:SENSe[1..n]:DATA:TELeom:OTN:OH:
OPU[1..n]:F:OVERhead?**

PSI11, selects PSI11 as overhead byte.

PSI12, selects PSI12 as overhead byte.

PSI13, selects PSI13 as overhead byte.

PSI14, selects PSI14 as overhead byte.

PSI15, selects PSI15 as overhead byte.

PSI16, selects PSI16 as overhead byte.

PSI17, selects PSI17 as overhead byte.

Example(s)

* SENS:DATA:TEL:OTN:OH:OPU1:F:OVER?

RES115 Returns the OPU overhead byte values.

See Also

* SOURce[1..n]:DATA:TELeom:OTN:OH:

OPU[1..n]:F:OVERhead

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: :TCM[1..n]:SAPI:B

Description	<p>This command allows editing the Source Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:SAPI:B<wsp><Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:F:TCM1:SAPI:B1 "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:ODU1:F:TCM1:SAPI:B1? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:TCM[1..n]:SAPI:B?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F
:TCM[1..n]:SAPI:B?**

Description	<p>This query returns the generated Source Access Point Identifier for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:SAPI:B? <wsp>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:TCM1:SAPI:B1 "EXFO OTU SAPI"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:TCM1:SAPI:B1? Returns "EXFO OTU SAPI"</p>
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:TCM[1..n]:SAPI:B

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: :TCM[1..n]:DAPI:B

Description	<p>This command allows editing the Destination Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:DAPI:B<wsp></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:F:TCM1:DAPI:B "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:ODU1:F:TCM1:DAPI:B? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:TCM1[1..n]:DAPI:B?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F
:TCM[1..n]:DAPI:B?**

Description	<p>This query returns the generated Destination Access Point Identifier for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:DAPI:B?<wsp>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:TCM1:DAPI:B "EXFO OTU DAPI"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:TCM1:DAPI:B? Returns "EXFO OTU DAPI"</p>
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:TCM1[1..n]:DAPI:B

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F :TCM[1..n]:OPSPec:B

Description	<p>This command allows editing the Operator Specific to be generated (TTI bytes 32 to 63) for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:OPSPec:B<wsp><Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:F:TCM1:OPSP:B "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:ODU1:F:TCM1:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:TCM[1..n]:OPSPec:B?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F
:TCM[1..n]:OPSPec:B?**

Description	<p>This query returns the generated Operator Specific for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:OPSPec:B?<wsp></code>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:TCM1:OPSP:B "EXFO OTU OPERATOR SPECIFIC"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:TCM1:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:TCM[1..n]:OPSPec:B</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: :PM:SAPI:B

Description	<p>This command allows editing the Source Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1f/2f Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: PM:SAPI:B<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:F:PM:SAPI:B "EXFO OTU SAPI" * SOUR:DATA:TEL:OTN:ODU1:F:PM:SAPI:B? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:PM:SAPI:B?</pre>

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

Description	<p>This command allows editing the Source Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1f/2f Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: PM:SAPI:B? <wsp></code>
Parameter(s)	None
Response Syntax	
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:PM:SAPI:B "EXFO OTU SAPI"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:PM:SAPI:B? Returns "EXFO OTU SAPI"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:PM:SAPI:B</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: :PM:DAPI:B

Description	<p>This command allows editing the Destination Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1f/2f Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: PM:DAPI:B<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the selected message for the instrument.</p>
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ODU1:F:PM:DAPI:B "EXFO OTU DAPI" * SOUR:DATA:TEL:OTN:ODU1:F:PM:DAPI:B? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:PM:DAPI:B?</pre>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:DAPI:B?

Description	<p>This query returns the generated Destination Access Point Identifier for ODU1f/2f Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:DAPI:B? <wsp>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:PM:DAPI:B "EXFO OTU DAPI"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:PM:DAPI:B? Returns "EXFO OTU DAPI"</p>
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:DAPI:B?

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: PM:OPSPec:B

Description	<p>This command allows editing the Operator Specific to be generated (TTI bytes 32 to 63) for ODU1f/2f Performance Monitoring.</p> <p>At *RST, this value is set to EXFO TCMi.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: PM:OPSPec:B<wsp><Message></p>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the selected message for the instrument.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:PM:OPSP:B "EXFO OTU OPERATOR SPECIFIC" * SOUR:DATA:TEL:OTN:ODU1:F:PM:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:PM:OPSPec:B?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:OPSPec:B?

Description	<p>This query returns the generated Destination Access Point Identifier for ODU1f/2f Performance Monitoring.</p> <p>At *RST, this value is set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:OPSPec:B?<wsp>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ODU1:F:PM:OPSP:B "EXFO OTU OPERATOR SPECIFIC"</p> <p>* SOUR:DATA:TEL:OTN:ODU1:F:PM:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</p>
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:PM:OPSPec:B

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:SAPI:EXPEcted

Description	<p>This command allows editing the expected Source Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:SAPI:EXPEcted <wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the expected message for the instrument.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:SAPI:EXP "EXFO OTU SAPI" * SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:SAPI:EXP? Returns "EXFO OTU SAPI"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:TIM * SENSe[1..n]:DATA:TELEcom:OTN:ODU1[1..n]:F:TCM[1..n]:TTI:SAPI:EXPEcted?</pre>

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
TCM[1..n]:TTI:SAPI:EXpected?**

Description	<p>This query returns the expected Source Access Point Identifier for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi SAPI.</p>
Syntax	<code>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:SAPI:EXpected?<wsp></code>
Parameter(s)	None
Response Syntax	<Message>

:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:SAPI:EXPeCted?

Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the expected message for the instrument.</p>
Example(s)	<ul style="list-style-type: none">* SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:TIM SAPI,ON* SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:SAPI:EXP "EXFO OTU SAPI"* SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:SAPI:EXP? Returns "EXFO OTU SAPI"
See Also	<ul style="list-style-type: none">* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:TIM* SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]:F:TCM[1..n]:TTI:SAPI:EXPeCted

:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:DAPI:EXPeCted

Description	<p>This command allows editing the expected Destination Access Point Identifier to be generated (TTI bytes 17 to 31) for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to EXFO TCMi DAPI.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:DAPI:EXPeCted<wsp> <Message></pre>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the expected message for the instrument.</p>
Example(s)	<pre>* SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:DAPI: EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:DAPI: EXP? Returns "EXFO OTU DAPI"</pre>
See Also	<pre>* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:TIM * SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]: F:TCM[1..n]:TTI:DAPI:EXPeCted?</pre>

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

Description This query returns the selected TCM level for non standard rates ODU1f/2f.

At *RST, this value is set to EXFO TCMi DAPI.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
TCM[1..n]:TTI:DAPI:EXPeCted? <wsp>

Parameter(s) None

Response Syntax <Message>

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F:
TCM[1..n]:TTI:DAPI:EXPeCted?**

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the expected message for the instrument.
Example(s)	* SENS:DATA:TEL:OTN:ODU1:F:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:ODU1:F:TTI:SAPI:EXP "EXFO OTU SAPI" * SENS:DATA:TEL:OTN:ODU1:F:TTI:SAPI:EXP? Returns "EXFO OTU SAPI"
See Also	* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TTI:TIM * SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]: F:TTI:DAPI:EXPeCted

**:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
TTI:SAPI:EXPEcted**

Description	<p>This command allows editing the expected Source Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1f/2f.</p> <p>At *RST, this value is set to EXFO ODU SAPI.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:SAPI:EXPEcted <wsp> <Message></p>
Parameter(s)	<p>Message: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the expected message for the instrument.</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:F:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:ODU1:F:TTI:DAPI:EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:ODU1:F:TTI:DAPI:EXP? Returns "EXFO OTU DAPI"</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:TIM * SENSe[1..n]:DATA:TELEcom:OTN:ODU1[1..n]: F:TTI:SAPI:EXPEcted?</p>

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

Description	<p>This command allows editing the expected Destination Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1f/2f.</p> <p>At *RST, this value is set to EXFO ODU DAPI.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:SAPI:EXpected? <wsp>
Parameter(s)	None
Response Syntax	<Message>

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

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the expected message for the instrument.
Example(s)	* SENS:DATA:TEL:OTN:ODU1:F:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:ODU1:F:TTI:DAPI:EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:ODU1:F:TTI:DAPI:EXP? Returns "EXFO OTU DAPI"
See Also	* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TTI:TIM * SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]: F:TTI:SAPI:EXPeCted

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

Description	<p>This command allows editing the expected Destination Access Point Identifier to be generated (TTI bytes 1 to 15) for ODU1f/2f.</p> <p>At *RST, this value is set to EXFO ODU DAPI.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:DAPI:EXPeCted<wsp><Message>
Parameter(s)	<p>Message:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Sets the expected message for the instrument.</p>
Example(s)	<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? Returns "EXFO OTU DAPI"</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:OTU[1..n]: TTI:TIM</p> <p>* SENSe[1..n]:DATA:TELEcom:OTN:OTU1[1..n]: TTI:DAPI:EXPeCted?</p>

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

Description	This query returns the expected Destination Access Point Identifier for ODU1f/2f. At *RST, this value is set to EXFO ODU DAPI.
Syntax	:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:DAPI:EXPEcted? <wsp>
Parameter(s)	None
Response Syntax	<Message>

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

Response(s)	Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the expected message for the instrument.
Example(s)	* SENS:DATA:TEL:OTN:ODU1:F:TTI:TIM DAPI,ON * SENS:DATA:TEL:OTN:ODU1:F:TTI:DAPI:EXP "EXFO OTU DAPI" * SENS:DATA:TEL:OTN:ODU1:F:TTI:DAPI:EXP? Returns "EXFO OTU DAPI"
See Also	* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TTI:TIM * SENSe[1..n]:DATA:TELecom:OTN:ODU1[1..n]: F:TTI:DAPI:EXPeCted

:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TTI:SAPI:B?

Description	<p>This command allows editing the expected Destination Access Point Identifier to be generated (TTI bytes 17 to 31) for ODU1f/2f.</p> <p>At *RST, this value is set to EXFO TCMi DAPI.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TTI:SAPI:B? <wsp></code>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ODU1:F:TTI:SAPI:B? Returns "EXFO OTU SAPI"</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
TTI:DAPI:B?**

Description	<p>This query returns the generated Destination Access Point Identifier for ODU1f/2f.</p> <p>At *RST, this value is set to EXFO TCMi DAPI.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:DAPI:B? <wsp>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ODU1:F:TTI:DAPI:B? Returns "EXFO OTU DAPI"</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
TTI:OPSPec:B?**

Description	<p>This query returns the generated Operator Specific for ODU1f/2f.</p> <p>At *RST, this value is set to EXFO TCMi OPERATOR SPECIFIC.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:OPSPec:B? <wsp></p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the selected message for the instrument.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ODU1:F:TTI:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
TCM[1..n]:TTI:SAPI:B?**

Description	<p>This query returns the generated Source Access Point Identifier for ODU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU SAPI".</p>
Syntax	FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:SAPI:B? <wsp>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	* FETC:DATA:TEL:OTN:ODU1:F:TCM1:TTI:SAPI: B? Returns "EXFO OTU SAPI"

:FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:DAPI:B?

Description	<p>This query returns the generated Destination Access Point Identifier for ODU1f/2f.</p> <p>At *RST, this value is set to "EXFO OTU DAPI".</p>
Syntax	<p>FETCh[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:DAPI:B? <wsp></p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Message></p>
Response(s)	<p>Message: The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element. Returns the selected message for the instrument.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ODU1:F:TCM1:TTI:DAPI: B? Returns "EXFO OTU DAPI"</p>

:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:OPSPec:B?

Description	<p>This query returns the generated Operator Specific for ODU1f/2f.</p> <p>At *RST, this value of OSPecific set to "EXFO OTU OPERATOR SPECIFIC".</p>
Syntax	:FETCh[1..n]:DATA:TELecom:OTN:ODU[1..n]:F:TCM[1..n]:TTI:OPSPec:B?<wsp>
Parameter(s)	None
Response Syntax	<Message>
Response(s)	<p>Message:</p> <p>The response data syntax for <Message> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the selected message for the instrument.</p>
Example(s)	* FETC:DATA:TEL:OTN:ODU1:F:TCM1:TTI:OPSP:B? Returns "EXFO OTU OPERATOR SPECIFIC"

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

Description This command enables or disables the state of TCM-Trace Identifier Mismatch (TIM) for the instrument for ODU1f/2f.

At *RST, this value is set to ON.

Syntax :SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
TCM[1..n]:TTI:TIM<wsp>SAPI|DAPI, <Set>

:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:TIM

Parameter(s)	<p>Etim:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SAPI DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:TIM SAPI,ON</p> <p>* SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:TIM? SAPI Return 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TCM[1..n]TTI:TIM?</p>

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

Description	<p>This query returns state of TCM-Trace Identifier Mismatch (TIM) for the instrument for ODU1f/2f.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TCM[1..n]:TTI:TIM? <wsp>SAPI DAPI</pre>
Parameter(s)	<p>Etim:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SAPI DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
Response Syntax	<Set>

**:SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F:
TCM[1..n]:TTI:TIM?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Trace Identifier Mismatch (TIM).
Example(s)	* SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:ODU1:F:TCM1:TTI:TIM? SAPI Return 1
See Also	* SENSe[1..n]:DATA:TELecom:OTN:ODU[1..n]:F: TCM[1..n]TTI:TIM

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

Description

This command enables or disables the state of Trace Identifier Mismatch (TIM) for the instrument for ODU1f/2f.

At *RST, this value is set to ON.

Syntax

:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
TTI:TIM<wsp>SAPI|DAPI, <Set>

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

Parameter(s)	<p>Etim:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SAPI DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p> <p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Enables or disables the Trace Identifier Mismatch (TIM).</p>
Example(s)	<p>* SENS:DATA:TEL:OTN:ODU1:F:TTI:TIM SAPI,ON</p> <p>* SENS:DATA:TEL:OTN:ODU1:F:TTI:TIM? SAPI</p> <p>Return 1</p>
See Also	<p>* SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:TTI:TIM?</p>

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

Description	<p>This query returns state of Trace Identifier Mismatch (TIM) for the instrument for ODU1f/2f.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:TIM? <wsp>SAPI DAPI</p>
Parameter(s)	<p>Etim:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: SAPI DAPI</p> <p>Enables or disables the TIM (Trace Identifier Mismatch) for the instrument.</p> <p>SAPI, selects the SAPI which allows editing of the SAPI (Source Access Point Identifier) message to be generated.</p> <p>DAPI, selects the DAPI which allows editing of the DAPI (Destination Access Point Identifier) message to be generated.</p>
Response Syntax	<p><Set></p>

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

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Trace Identifier Mismatch (TIM).
Example(s)	* SENS:DATA:TEL:OTN:ODU1:F:TTI:TIM SAPI,ON * SENS:DATA:TEL:OTN:ODU1:F:TTI:TIM? SAPI Return 1
See Also	* SENSe[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: TTI:TIM

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: :CONFig:TCM[1..n]

Description	<p>This command enables or disables the configuration of TCM for ODU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: CONFig:TCM[1..n] <wsp> <LEVEL></pre>
Parameter(s)	<p>LEVEL:</p> <p>The program data syntax for the parameter is defined as a <BOOLEAN PROGRAM DATA> element.</p> <p>Enables or disables the configuration of TCM.</p>
Example(s)	<pre>* SOURce:DATA:TEL:OTN:ODU1:F:CONF:TCM1 ON * SOURce:DATA:TEL:OTN:ODU1:F:CONF:TCM1? Returns 1</pre>
See Also	<pre>* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:CONF:TCM[1..n]?</pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F:
:CONFig:TCM[1..n]?**

Description	<p>This query returns the status of the configuration of TCM for ODU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F: CONFig:TCM[1..n]? <wsp></p>
Parameter(s)	<p>Level: The program data syntax for the parameter is defined as a <BOOLEAN NUMERIC PROGRAM DATA> element. Enables or disables the configuration of TCM.</p>
Response Syntax	<p><SET></p>

:SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]:F :CONFig:TCM[1..n]?

Response(s)	SET: The response data syntax for <SET> is defined as a <NR1 NUMERIC RESPONSE DATA> element.
Example(s)	* SOURce:DATA:TEL:OTN:ODU1:F:CONF:TCM1 ON * SOURce:DATA:TEL:OTN:ODU1:F:CONF:TCM1? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ODU[1..n]: F:CONF:TCM[1..n]

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:TYPE****Description**

This command selects the type of Optical Data Unit (ODU) alarm for ODU1f/2f of the TCM level.

At *RST, this value is set to TLTC.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:TYPE<wsp>TLTC|TBDI|
TIQE|TBIAE

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:TYPE**

Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TLTC TBDI TIQE TBIAE</p> <p>Selects the type of ODU (Optical Data Unit) alarm.</p> <p>TLTC, selects TCMi - Loss of Tandem Connection as the type of ODU (Optical Data Unit) alarm.</p> <p>TBDI, selects TCMi - Backward Defect Indication as the type of ODU (Optical Data Unit) alarm.</p> <p>TIQE, selects TCMi - Incoming Alignment Error as the type of ODU (Optical Data Unit) alarm.</p> <p>TBIAE, selects TCMi - Backward Incoming Alignment Error as the type of ODU (Optical Data Unit) alarm.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TCM1:TYPE TLTC</p> <p>* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TCM1:TYPE? Returns TLTC</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:F:TCM[1..n]:TYPE?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:TYPE?**

Description	<p>This query returns the type of Optical Data Unit (ODU) alarm for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to TLTC.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TCM[1..n]:TYPE? <wsp></pre>
Parameter(s)	None
Response Syntax	<Alarm>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:TYPE?**

Response(s)

Alarm:

The response data syntax for <Alarm> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the type of Optical Data Unit (ODU) alarm.

TLTC, TCMi - Loss of Tandem Connection is selected as the type of ODU (Optical Data Unit) alarm.

TBDI, TCMi - Backward Defect Indication is selected as the type of ODU (Optical Data Unit) alarm.

TIAE, TCMi - Incoming Alignment Error is selected as the type of ODU (Optical Data Unit) alarm.

TBIAE, TCMi - Backward Incoming Alignment Error is selected as the type of ODU (Optical Data Unit) alarm.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TCM1:
TYPE TLTC

* SOUR:DATA:TEL:OTN:ALAR:ODU1:F:TCM1:
TYPE? Returns TLTC

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:TYPE

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:F:
ODUTCM[1..n]**

Description	<p>This command enables or disables the status of Optical Channel Data Unit (ODU) alarm generation for ODU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:F: ODUTCM[1..n] <wsp> <Level>, <Set></p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:F:
ODUTCM[1..n]**

Parameter(s)	<p>Level: The program data syntax for the parameter is defined as a <NRI NUMERIC PROGRAM DATA> element. Enables or disables the ODU (Optical Data Unit) alarm generation.</p> <p>Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. Enables or disables the Trace Identifier Mismatch (TIM).</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ALAR:F:ODUTCM1 1,ON</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:F:TCM[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:F:TCM[1..n]?</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:F:
ODUTCM[1..n]?**

Description	<p>This query returns the status of Optical Data Unit (ODU) alarm generation for ODU1f/2f.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:F: ODUTCM[1..n]?<wsp><Level></p>
Parameter(s)	<p>Level:</p> <p>The program data syntax for the first parameter is defined as a <NRI NUMERIC PROGRAM DATA> element.</p> <p>Selects the TCM level for alarm type.</p>
Response Syntax	<p><Set></p>

:SOURce[1..n]:DATA:TELEcom:OTN:ALARm:F:ODUTCM[1..n]?

Response(s)

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Optical Data Unit (ODU) alarm generation.

Example(s)

* SOUR:DATA:TEL:OTN:ALAR:F:ODUTCM2 1,ON
* SOUR:DATA:TEL:OTN:ALAR:F:ODUTCM2? 1
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:TYPE
* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:HISTory?****Description**

This query returns the history status of Optical Data Unit (ODU) alarm for ODU1f/2f of the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:HISTory?<wsp>TLTC |
TBDI | TTIM | TBIAE | TIAE

**:FETCh[1..n]:DATA:TELeom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:HISTory?**

Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TLTC TBDI TTIM TBIAE TIAE.</p> <p>Selects the history status of Optical Data Unit (ODU) alarm for the TCM level.</p> <p>TLTC, selects TCMi - Loss of Tandem Connection as the history status of Optical Data Unit (ODU) alarm.</p> <p>TBDI, selects TCMi - Backward Defect Indication as the history status of Optical Data Unit (ODU) alarm.</p> <p>TTIM, selects TCMi - Trace Identifier Mismatch as the history status of Optical Data Unit (ODU) alarm.</p> <p>TBIAE, selects TCMi - Backward Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.</p> <p>TIAE, selects TCMi - Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.</p>
Response Syntax	<History>

**:FETCh[1..n]:DATA:TELeom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <Message> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<p>* FETC:DATA:TEL:OTN:ALAR:ODU1:F:TCM1:HIST? TLTC Returns the ALAR history.</p>
See Also	<p>* SOURce[1..n]:DATA:TELeom:OTN:ALARm:ODU[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELeom:OTN:ALARm:ODU[1..n]</p>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TCM[1..n]:SEConds?

Description

This query returns the number of seconds within which Optical Data Unit (ODU) alarm occurred for ODU1f/2f of the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:SEConds? <wsp>TLTC |
TBDI | TTIM | TBIAE | TIAE

:FETCh[1..n]:DATA:TELeom:OTN:ALARm: ODU[1..n]:F:TCM[1..n]:SEConds?

Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TLTC TBDI TTIM TBIAE TIAE.</p> <p>Selects the number of seconds within which Optical Data Unit (ODU) alarm occurred for the TCM level.</p> <p>TLTC, selects TCMi - Loss of Tandem Connection as the number of seconds within which Optical Data Unit (ODU) alarm occurred.</p> <p>TBDI, selects TCMi - Backward Defect Indication as the number of seconds within which Optical Data Unit (ODU) alarm occurred.</p> <p>TTIM, selects TCMi - Trace Identifier Mismatch as the number of seconds within which Optical Data Unit (ODU) alarm occurred.</p> <p>TBIAE, selects TCMi - Backward Incoming Alignment Error as the number of seconds within which Optical Data Unit (ODU) alarm occurred.</p> <p>TIAE, selects TCMi - Incoming Alignment Error as the number of seconds within which Optical Data Unit (ODU) alarm occurred.</p>
Response Syntax	<Seconds>

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm: ODU[1..n]:F:TCM[1..n]:SEConds?

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Data Unit (ODU) alarm.
Example(s)	* FETC:DATA:TEL:OTN:ALAR:ODU1:F:TCM1:SEC? TLTC Returns the ALAR seconds.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ALARm:ODU[1..n]

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:CURRent?****Description**

This query returns the current status of Optical Data Unit (ODU) alarm occurred for ODU1f/2f of the TCM level.

At *RST, this value is device dependent.

Syntax

:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:CURRent? <wsp> TLTC
|TBDI|TTIM|TBIAE|TIAE

**:FETCh[1..n]:DATA:TELEcom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:CURRent?**

Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TLTC TBDI TTIM TBIAE TIAE</p> <p>Selects the current status of Optical Data Unit (ODU) alarm for the TCM level.</p> <p>TLTC, selects TCMi - Loss of Tandem Connection as the history status of Optical Data Unit (ODU) alarm.</p> <p>TBDI, selects TCMi - Backward Defect Indication as the history status of Optical Data Unit (ODU) alarm.</p> <p>TTIM, selects TCMi - Trace Identifier Mismatch as the history status of Optical Data Unit (ODU) alarm.</p> <p>TBIAE, selects TCMi - Backward Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.</p> <p>TIAE, selects TCMi - Incoming Alignment Error as the history status of Optical Data Unit (ODU) alarm.</p>
Response Syntax	<Current>

**:FETCh[1..n]:DATA:TELeom:OTN:ALARm:
ODU[1..n]:F:TCM[1..n]:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> 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>
Example(s)	<p>* FETC:DATA:TEL:OTN:ALAR:ODU1:F:TCM1:CURR? TLTC Returns the ALAR current.</p>
See Also	<p>* SOURce[1..n]:DATA:TELeom:OTN:ALARm:ODU[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELeom:OTN:ALARm:ODU[1..n]</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:EROR:
ODU[1..n]:F:TCM[1..n]:HISTory?**

Description	<p>This query returns the history status of Optical Data Unit (ODU) error.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:EROR: ODU[1..n]:F:TCM[1..n]:HISTory?<wsp>TBIP8 TBEI</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI</p> <p>Selects the history status of Optical Data Unit (ODU) error for the TCM level.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.</p>
Response Syntax	<p><History></p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:HISTory?**

Response(s)	<p>History:</p> <p>The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the history status of Optical Data Unit (ODU) 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>
Example(s)	<p>* FETC:DATA:TEL:OTN:ERR:ODU1:F:TCM1:HIST? TBIP8 Returns the Error history.</p>
See Also	<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]:AMOUNT</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:INJect</p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:TCM[1..n]:SECOnds?**

Description This query returns the number of seconds within which Optical Data Unit (ODU) error occurred for the TCM level.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:TCM[1..n]:SECOnds? <wsp>TBIP8|
TBEI

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8|TBEI
Selects the number of seconds within which Optical Data Unit (ODU) error occurred for the TCM level.
TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.
TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.

Response Syntax <Seconds>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:SEConds?**

Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of seconds of Optical Data Unit (ODU) error.
Example(s)	* FETC:DATA:TEL:OTN:ERR:ODU1:F:TCM1: SEC? TBIP8 Returns the number of errored seconds.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AMOUNT * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:INJect

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:F:TCM[1..n]:CURREnt?

Description	<p>This query returns the current status of Optical Data Unit (ODU) error for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<pre>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:F:TCM[1..n]:CURREnt?<wsp>TBIP8 TBEI</pre>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI</p> <p>Selects the current status of Optical Data Unit (ODU) error for the TCM level.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.</p>
Response Syntax	<pre><Current></pre>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:CURRent?**

Response(s)	<p>Current:</p> <p>The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the current status of Optical Data Unit (ODU) 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>
Example(s)	<p>* FETC:DATA:TEL:OTN:ERR:ODU1:F:TCM1:CURR? TBIP8 Returns the current error status.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOUNT* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:INJect

:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:F:TCM[1..n]:COUNT?

Description This query returns the count of Optical Data Unit (ODU) error for ODU1f/2f of the TCM level.

At *RST, this value is device dependent.

Syntax :FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:TCM[1..n]:COUNT?<wsp>TBIP8|
TBEI

Parameter(s) Error:
The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.
The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8|TBEI
Selects the count of Optical Data Unit (ODU) error for the TCM level.
TBIP8, selects the TCMi - Bit Interleave Parity-8 as the count of Optical Data Unit (ODU) error.
TBEI, selects the TCMi - Backward Error Indication as the count of Optical Data Unit (ODU) error.

Response Syntax <Count>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:COUNT?**

Response(s)	Count: The response data syntax for <Count> is defined as a <NR2 NUMERIC RESPONSE DATA> element. Returns the count of Optical Data Unit (ODU) error.
Example(s)	* FETC:DATA:TEL:OTN:ERR:ODU1:F:TCM1: COUN? TBIP8 Returns the error count.
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:MANual:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:AMOUNT * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:INJect

**:FETCh[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:TCM[1..n]:RATE?**

Description	<p>This query returns the current rate of Optical Data Unit (ODU) error for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:F:TCM[1..n]:RATE? <wsp>TBIP8 TBEI</p>
Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI</p> <p>Selects the current rate of Optical Data Unit (ODU) error for ODU1f/2f of the TCM level.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the current rate of Optical Data Unit (ODU) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the current rate of Optical Data Unit (ODU) error.</p>
Response Syntax	<p><Rate></p>

**:FETCh[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the current rate of Optical Data Unit (ODU) error.</p>
Example(s)	<p>* FETC:DATA:TEL:OTN:ERR:ODU1:F:TCM1:RATE? TBIP8 Returns the error rate.</p>
See Also	<ul style="list-style-type: none">* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU[1..n]:MANual:TYPE* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:AMOUNT* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:INJect

:SOURce[1..n]:DATA:TELEcom:OTN:ERROr: ODU[1..n]:F:AUTomated:TCM[1..n]:TYPE

Description

This command selects the type of Optical Data Unit (ODU) error for ODU1f/2f automated injection of the TCM level.

At *RST, this value is set to BERRor.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERROr:
ODU[1..n]:F:AUTomated:TCM[1..n]:TYPE<wsp>
TBIP8|TBEI

**:SOURce[1..n]:DATA:TELecom:OTN:ERRor:
ODU[1..n]:F:AUTomated:TCM[1..n]:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI</p> <p>Returns the type of Optical Data Unit (ODU) error for the automated injection.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of Optical Data Unit (ODU) error for automated injection.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of Optical Data Unit (ODU) error for automated injection.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:TYPE? Returns TBIP8</p>
See Also	<p>* SOURce[1..n]:DATA:TELecom:OTN:ERRor:ODU1:F:AUTomated:TCM[1..n]TYPE?</p>

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:F:AUTomated:TCM[1..n]:TYPE?

Description	<p>This query returns the type of Optical Data Unit (ODU) error for ODU1f/2f automated injection of the TCM level.</p> <p>At *RST, this value is set to TBIP8.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:F:AUTomated:TCM[1..n]:TYPE? <wsp></p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Error></p>

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:F:AUTomated:TCM[1..n]:TYPE?****Response(s)**

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element. Returns the type of Optical Data Unit (ODU) error for the automated injection.

TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of Optical Data Unit (ODU) error for automated injection.

TBEI, selects the TCMi - Backward Error Indication as the type of Optical Data Unit (ODU) error for automated injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:
TCM1TYPE TBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:
TYPE? Returns TBIP8

See Also

* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU1:F:AUTomated:TCM[1..n]TYPE?

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TCM[1..n]:RATE

Description

This command selects the rate of Optical Data Unit (ODU) error for ODU1f/2f automated injection of the TCM level.

At *RST, this value is device dependent.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:TCM[1..n]:RATE<wsp>
MAXimum|MINimum

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TCM[1..n]:RATE

Parameter(s)	<p>Rate:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum</p> <p>Sets the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:RATE? Returns 1.0E-09</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:F:AUTomated:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:F:AUTomated:TCM[1..n]:RATE?</p>

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:F:AUTomated:TCM[1..n]:RATE?

Description	<p>This query returns the rate of Optical Data Unit (ODU) error for ODU1f/2f automated injection of the TCM level.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:F:AUTomated:TCM[1..n]:RATE? <wsp>MAXimum MINimum</p>
Parameter(s)	<p>None:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum</p> <p>Sets the injection rate for the selected Optical Data Unit (ODU) error.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the injected rate will be returned.</p>
Response Syntax	<p><Rate></p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:TCM[1..n]:RATE?**

Response(s)	<p>Rate:</p> <p>The response data syntax for <Rate> is defined as a <NR3 NUMERIC RESPONSE DATA> element.</p> <p>Returns the injection rate for the selected Optical Data Unit (ODU) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:RATE 1.0E-09</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:RATE? Returns 1.0E-09</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:F:AUTomated:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:F:AUTomated:TCM[1..n]:RATE</p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:AUTomated**

Description

This command enables or disables the selected automated Optical Data Unit (ODU) error at the rate specified or continuously for ODU1f/2f of the TCM level.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:AUTomated<wsp>
<Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:AUTomated**

Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. Enables or disables the automated Optical Data Unit (ODU) error injection.
Example(s)	* OUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1: TYPE OBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1: RATE 1.0E-09 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AUT ON * SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AUT? Returns 1
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU1:F:AUTomated:TCM[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU1:F:AUTomated:TCM[1..n]:RATE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU1:F:TCM[1..n]:AUTomated?

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:TCM[1..n]:AUTomated?**

Description This query returns the status of automated Optical Data Unit (ODU) error injection for ODU1f/2f of the TCM level.

At *RST, this value is set to OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:AUTomated? <wsp>

Parameter(s) None

Response Syntax <Set>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:TCM[1..n]:AUTomated?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of automated Optical Data Unit (ODU) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:
TYPE OBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:
RATE 1.0E-09

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AUT
ON

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AUT?
Returns 1

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:F:AUTomated:TCM[1..n]:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:F:AUTomated:TCM[1..n]:RATE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU1:F:TCM[1..n]:AUTomated

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU [1..n]:F:AUTomated:TCM[1..n]:CONTInuous?

Description

This command enables or disables the continuously rate of automated Optical Data Unit (ODU) error injection for non standard rates ODU1f/2f of the TCM level.

At *RST, this value is set to OFF.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:AUTomated:TCM[1..n]:CONTInuous
<wsp> <Set>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU [1..n]:F:AUTomated:TCM[1..n]:CONTInuous?

Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element. Enables or disables the continuous rate of automated Optical Data Unit (ODU) error injection.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:CONT ON</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:CONT? Returns 1</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AUT ON</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:F:AUTomated:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:F:TCM[1..n]:AUTomated</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:F:AUTomated:TCM[1..n]:CONTInuous?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU [1..n]:F:AUTomated:TCM[1..n]:CONTInuous?

Description	<p>This query returns the status of continuously rate of automated Optical Data Unit (ODU) error injection for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:AUTomated:TCM[1..n]: CONTInuous?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Set></p>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU
[1..n]:F:AUTomated:TCM[1..n]:CONTInuous?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of continuously rate of automated Optical Data Unit (ODU) error injection.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:TYPE TBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:CONT ON

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:AUT:TCM1:CONT? Returns 1

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AUT ON

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:F:AUTomated:TCM[1..n]:TYPE

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:F:TCM[1..n]:AUTomated

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU1:F:AUTomated:TCM[1..n]:CONTInuous?

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:F:MANual:TCM[1..n]:TYPE

Description

This command selects the manual type of Optical Data Unit (ODU) error for ODU1f/2f of the TCM level.

At *RST, this value is set to OBIP8.

Syntax

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:F:MANual:TCM[1..n]:TYPE<wsp>
TBIP8|TBEI

**:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:
ODU[1..n]:F:MANual:TCM[1..n]:TYPE**

Parameter(s)	<p>Error:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TBIP8 TBEI.</p> <p>Selects the manual type of Optical Data Unit (ODU) error for ODU1f/2f TCM level.</p> <p>TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.</p> <p>TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TCM1:TYPE? Returns TBIP8</p>
See Also	<p>* SOURCE[1..n]:DATA:TELEcom:OTN:ERROR:ODU[1..n]:F:MANual:TCM[1..n]TYPE?</p>

:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:F:MANual:TCM[1..n]:TYPE?

Description	<p>This query returns the manual type of Optical Data Unit (ODU) error for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to OBIP8.</p>
Syntax	<code>:SOURCE[1..n]:DATA:TELEcom:OTN:ERROR: ODU[1..n]:F:MANual:TCM[1..n]:TYPE?<wsp></code>
Parameter(s)	None
Response Syntax	<Error>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:MANual:TCM[1..n]:TYPE?****Response(s)**

Error:

The response data syntax for <Error> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the type of Optical Data Unit (ODU) error.

TBIP8, selects the TCMi - Bit Interleave Parity-8 as the type of ODU (Optical Data Unit) error.

TBEI, selects the TCMi - Backward Error Indication as the type of ODU (Optical Data Unit) error.

Example(s)

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TCM1:TYPE TBIP8

* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TCM1:TYPE? Returns TBIP8

See Also

* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:MANual:TCM[1..n]TYPE

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:AMOUnt

Description

This command sets the amount of Optical Data Unit (ODU) error to inject for ODU1f/2f of the TCM level.

At *RST, this value is set to 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:AMOUnt <wsp>
MAXimum | MINimum

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:AMOUNT

Parameter(s)	<p>Amount:</p> <p>The program data syntax for the parameter is defined as a <numeric_value> element.</p> <p>The allowed <numeric_value> elements for this parameter are: MAXimum MINimum.</p> <p>Sets the amount of ODU (Optical Data Unit) error. Choices are 1 through 50. The default setting is 1.</p> <p>MAXimum is used to retrieve the instrument's greatest supported value.</p> <p>MINimum is used to retrieve the instrument's smallest supported value.</p>
Example(s)	<p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TCM1:TYPE TBIP8</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AMO 15</p> <p>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AMO? Returns 15</p>
See Also	<p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:MANual:TCM[1..n]:TYPE</p> <p>* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:TCM[1..n]:AMOUNT?</p>

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:AMOUnt?

Description	<p>This query returns the amount of Optical Data Unit (ODU) error to inject for ODU1f/2f of the TCM level.</p> <p>At *RST, this value is set to 1.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:F:TCM[1..n]:AMOUnt?<wsp> MAXimum MINimum</pre>
Parameter(s)	<p>None</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MAXimum MINimum.</p> <p>Sets the amount of ODU (Optical Data Unit) error. MAXimum is used to retrieve the instrument's greatest supported value. MINimum is used to retrieve the instrument's smallest supported value.</p> <p>This parameter is optional. If no token is specified, the current amount will be returned.</p>
Response Syntax	<pre><Amount></pre>

**:SOURce[1..n]:DATA:TELEcom:OTN:ERRor:
ODU[1..n]:F:TCM[1..n]:AMOut?**

Response(s)	Amount: The response data syntax for <Amount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the amount of Optical Data Unit (ODU) error.
Example(s)	* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TCM1:TYPE TBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AMO? Returns 15
See Also	* SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:MANual:TCM[1..n]:TYPE * SOURce[1..n]:DATA:TELEcom:OTN:ERRor:ODU[1..n]:F:TCM[1..n]:AMOut?

:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:INJect

Description	<p>This query returns the amount of Optical Channel Data Unit (ODU) error to inject for ODU1f/2f of the TCM level.</p> <p>This command is an event and has no associated *RST condition or query form.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:OTN:ERRor: ODU[1..n]:E:TCM[1..n]:INJect<wsp></pre>
Parameter(s)	None
Example(s)	<pre>* SOUR:DATA:TEL:OTN:ERR:ODU1:F:MAN:TCM1: TYPE TBIP8 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:AMO 15 * SOUR:DATA:TEL:OTN:ERR:ODU1:F:TCM1:INJ</pre>

**:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:
WAVelength****Description**

This command sets the Optical Wavelength value.

At *RST, this value is set to 1310.

Syntax

:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:
WAVelength<wsp>MAXimum|MINimum

Parameter(s)

Value:

The program data syntax for the parameter is defined as a <numeric_value> element.

The allowed <numeric_value> elements for this parameter are: MAXimum|MINimum.

Sets the Optical Wavelength Value.

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

Description This query returns the Optical Wavelength value.

At *RST, this value is set to 1310.

Syntax :SOURce[1..n]:DATA:TELEcom:OPTical:PORT:
WAVelength? <wsp> MAXimum | MINimum

Parameter(s) The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.

The allowed <CHARACTER PROGRAM DATA> elements for this parameter are:
MAXimum | MINimum.

This parameter is optional. If no token is specified, the injected rate will be returned.

Response Syntax <Value>

Response(s) Value:

The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element.

Returns the Optical Wavelength value.

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

Description	This command enables or disables the Invert Polarity.
Syntax	:SOURce[1..n]:DATA:TELEcom:OPTical:PORT: IPOPolarity<wsp><Set>
Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. Enables or disables the Invert Polarity.
Example(s)	SOUR:DATA:TEL:OPT:PORT:WAV
See Also	SOURce[1..n]:DATA:TELEcom:OPTical:PORT: WAVElength

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

Description	This query returns the status for Invert Polarity.
Syntax	:SOURce[1..n]:DATA:TELEcom:OPTical:PORT:IPOLarity?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 Numeric Response Data> element. Returns the status of Invert Polarity.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:SMODE****Description**

This command sets the switching mode for RS-MS. Available on both TX and RX tabs.

At *RST, this value is Linear.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:SMODE<wsp>LINEar|RING

Parameter(s)

Mode:

The program data syntax for the parameter is defined as a <Character Program Data> element.

The allowed <Character Program Data> elements for this parameter are: LINEar|RING.

Command sets the switching mode for RS-MS.

LINEar, selects Linear as Switching Mode.

RING, selects Ring as Switching Mode.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:SMODE?**

Description	<p>This query returns the switching mode for RS-MS. Available on both TX and RX tabs.</p> <p>At *RST, this value is Linear.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:SMODE?</p>
Parameter(s)	<p>None.</p>
Response Syntax	<p><Type></p>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character Response Data> 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>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:OVERwrite**

Description	<p>This command enables or disables the overwrite, which allows the activation and deactivation of APS.</p> <p>At *RST, this value is OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:OVERwrite<wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Enables or disables overwrite.</p>

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:OVERwrite?

Description	This query returns the overwrite status. At *RST, this value is OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:OVERwrite?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 Numeric Response Data> element. Returns the overwrite status.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:REQuest**

Description This command selects the K1 Request for Linear Switching Mode.

At *RST, this value is No Request (0000).

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K: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:LINEar:REQuest**

Parameter(s)	Type:
	<p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> 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, selects the No Request (0000) request type.</p> <p>DNREVERT0001, selects the Do Not Revert (0001) request type.</p> <p>RREQUEST0010, selects the Reverse Request (0010) request type.</p> <p>UNUSED0011, selects the unused (0011) request type.</p> <p>EXERCISER0100, selects the Exerciser (0100) request type.</p> <p>UNUSED0101, selects the Unused (0101) request type.</p> <p>WTRESTORE0110, selects the Wait to restore (0110) request type.</p> <p>UNUSED0111, selects the Unused (0111) request type.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:REQuest**

Parameter(s)	UNUSED1001, selects the Unused (1001) request type.
	SDLOW1010, selects the Signal Degrade (1010) request type.
	SDHIGH1011, selects the Signal Degrade - High Priority (1011) request type.
	SFLOW1100, selects the Signal Fail - Low Priority (1100) request type.
	SFHIGH1101, selects the Signal Fail - High Priority (1101) request type.
	FSWITCH1110, selects the Forced Switch (1110) request type.
	MSWITCH1000, selects the Manual Switch request type.
	LPROTECTION1111, selects the Lockout of Protection (1111) request type.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:REQuest?**

Description This query returns the K1 request for Linear Switching Mode.

At *RST, this value is No Request (0000).

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:REQuest?

Parameter(s) None

Response Syntax <Type>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:REQuest?****Response(s)**

Type:

The response data syntax for <Type> is defined as a <Character Response Data> element.

Returns the K1 request for Linear Switching Mode.

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.

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:REQuest?

Response(s)	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.
--------------------	---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:REQuest****Description**

This command selects the K1 Request for Ring Switching Mode.

At *RST, this value is No Request (0000).

Syntax

```
:SOURce[1..n]:DATA:TELEcom:SDHSonet:  
ADVanced:APS:K: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:RING:REQuest

Parameter(s)	Type:
	<p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> 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 - Selects the No Request (0000) request type.</p> <p>DNREVERT0001 - Selects the Do Not Revert (0001) request type.</p> <p>UNUSED1001 - Selects the Unused (1001) request type.</p> <p>SDLOW1010 - Selects the Signal Degrade (1010) request type.</p> <p>SDHIGH1011 - Selects the Signal Degrade - High Priority (1011) request type.</p> <p>SFLOW1100 - Selects the Signal Fail - Low Priority (1100) request type.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:REQuest**

Parameter(s)	
	SFHIGH1101 - Selects the Signal Fail - High Priority (1101) request type.
	FSWITCH1110 - Selects the Forced Switch (1110) request type.
	LPROTECTION1111 - Selects the Lockout of Protection (1111) request type.

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:REQuest?

Description This query returns the K1 request for Ring Switching Mode.

At *RST, this value is No Request (0000).

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:REQuest?

Parameter(s) None

Response Syntax <Type>

**:SOURCE[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:REQuest?****Response(s)**

Type:

The response data syntax for <Type> is defined as a <Character Response Data> element.

Returns the Ring 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.

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:REQuest?

Response(s)	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.
--------------------	---

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:CHANnel**

Description	<p>This command selects the K1 Channel for Linear Switching mode.</p> <p>At *RST, this value is 0 - Null.</p>
Syntax	<pre>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:CHANnel<wsp><Channel></pre>
Parameter(s)	<p>Channel:</p> <p>The program data syntax for the parameter is defined as a <Numeric Value> element.</p> <p>Sets the K1 Channel for Linear Switching.</p> <p>Range is 0 to 15.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:CHANnel?**

Description	This query returns the K1 Channel for Linear Switching Mode. At *RST, this value is 0 - Null.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:CHANnel?
Parameter(s)	None
Response Syntax	<Value>
Response(s)	Value: The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element. Returns the K1 Channel for Linear Switching Mode.

:SOURce[1..n]:DATA:TELEcom:SDHSONet: ADVanced:APS:K:PCHannel

Description This command sets the K2 Protected Channel for Linear Switching Mode.

At *RST, this value is 0 - Null.

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSONet:
ADVanced:APS:K:PCHannel<wsp> <Channel>

Parameter(s) Channel:
The program data syntax for the parameter is defined as a <Numeric Value> element.
Selects the K2 Protected Channel for Linear Switching.
Range is 0 to 15.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:PCHannel?**

Description	<p>This query returns the K2 Protected Channel for Linear Switching mode.</p> <p>At *RST, this value is 0 - Null.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:PCHannel?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Value></p>
Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element.</p> <p>Returns the K2 Protected Channel for Linear Switching mode.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:ARCHitecture****Description**

This command sets the K2 Architecture for Linear Switching mode.

At *RST, this value is 1 + 1.

Syntax

:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:ARCHitecture <wsp>
1TO1 | 1TON

Parameter(s)

Type:

The program data syntax for the parameter is defined as a <Character Program Data> element.

The allowed <Character Program Data> elements for this parameter are: 1TO1 | 1TON.

Selects the K2 Architecture for Linear Switching.

1TO1, Selects the 1+1 Architecture.

1TON, Selects the 1:N Architecture.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:ARCHitecture?**

Description This query returns the K2 Architecture for Linear switching mode.

At *RST, this value is 1 + 1.

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:ARCHitecture?

Parameter(s) None

Response Syntax <Type>

Response(s) Type:
The response data syntax for <Type> is defined as a <Character Response Data> element.
Returns the K2 Architecture for Linear switching mode. Possible values are: 1TO1 | 1TON.
1TO1, Selects the 1 + 1 Architecture.
1TON, Selects the 1:N Architecture.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:OMODE**

Description

This command sets the K2 Operation mode for Linear Switching mode.

At *RST, this value is Reserved (000).

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:OMODE

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
 ADVanced:APS:K:LINEar:OMODE <wsp>
 RESERVED000 | RESERVED001 | RESERVED010 |
 RESERVED011 | UNI100 | BID101 | MSRD110
 | MSAIS111

:SOURCE[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:OMODE

Parameter(s)	<p>Operation Mode:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: RESERVED000 RESERVED001 RESERVED010 RESERVED011 UNI100 BID101 MSRD110 MSAIS111.</p> <p>Selects the K2 Operation Mode for Linear Switching Mode.</p> <p>RESERVED000, Selects the Reserved (000) Operation Mode.</p> <p>RESERVED001, Selects the Reserved (001) Operation Mode.</p> <p>RESERVED010, Selects the Reserved (010) Operation Mode.</p> <p>RESERVED011, Selects the Reserved (011) Operation Mode.</p> <p>UNI100, selects the Unidirectional (100) Operatin Mode.</p> <p>BID101, selects the Bidirectional (101) Operation Mode.</p> <p>MSRD110, selects the MS-RDI (110) Operation Mode.</p> <p>MSAIS111, Selects the MS-AIS (111) Operation Mode.</p>
---------------------	---

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:OMODE?

Description This query returns the K2 Operation mode for Linear Switching mode.

At *RST, this value is Reserved (000).

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:OMODE?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:OMODE?**

Parameter(s)	None
Response Syntax	<Type>
Response(s)	Type: The response data syntax for <Type> is defined as a <Character Response Data> element. Returns the K2 Operation mode for Linear Switching mode. RESERVED000 - Reserved (000) Operation Mode is selected. RESERVED001 - Reserved (001) Operation Mode is selected. RESERVED010 - Reserved (010) Operation Mode is selected. RESERVED011 - Reserved (011) Operation Mode is selected. UNI100 - Unidirectional (100) Operatin Mode is selected. BID101 - Bidirectional (101) Operation Mode is selected. MSRDI110 - MS-RDI (110) Operation Mode is selected. MSAIS111 - MS-AIS (111) Operation Mode is selected.

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:OMODE

Description

This command sets the Operation mode for Ring Switching mode.

At *RST, this value is Idle (000).

Syntax

```
:SOURce[1..n]:DATA:TELEcom:SDHSonet:  
ADVanced:APS:K:RING:OMODE<wsp>  
MSRDI110|MSAIS111|IDLE000|BRIDGED001|  
BSWITCHED010|ETPROTECTION011|  
RESERVED100|RESERVED101
```


**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:OMODE****Parameter(s)**

Operation Mode:

The program data syntax for the parameter is defined as a <Character Program Data> element.

The allowed <Character Program Data> elements for this parameter are:

MSRDI110|MSAIS111|IDLE000|BRIDGED001|
BSWITCHED010|ETPROTECTION011|
RESERVED100|RESERVED101.

Selects the K2 Operation Mode for Linear and Ring Switching.

MSRDI110 - Selects the MS-RDI (110) Operation Mode.

MSAIS111 - Selects the MS-AIS (111) Operation Mode.

IDLE000 - Selects the Idle (000) Operation Mode.

BRIDGED001 - Selects Bridge (001) Operation Mode.

BSWITCHED010 - Select Bridged and Switched (010) Operation Mode.

ETPROTECTION011 - Selects Extra Traffic - Protection (011) Operation Mode.

RESERVED100 - Selects Reserved (100) Operation Mode.

RESERVED101 - Selects Reserved (101) Operation Mode.

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:OMODE?

Description

This query returns the Operation mode for Ring Switching mode.

At *RST, this value is Idle (000).

Syntax

:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:OMODE?

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:OMODE?**

Parameter(s)	None
Response Syntax	<Type>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character Response Data> element.</p> <p>Returns the K2 Operation mode for Ring Switching mode.</p> <p>MSRDI110 - MS-RDI (110) Operation Mode is selected.</p> <p>MSAIS111 - MS-AIS (111) Operation Mode is selected.</p> <p>IDLE000 - Idle (000) Operation Mode is selected.</p> <p>BRIDGED001 - Bridge (001) Operation Mode is selected.</p> <p>BSWITCHED010 - Bridged and Switched (010) Operation Mode is selected.</p> <p>ETPROTECTION011 - Extra Traffic - Protection (011) Operation Mode is selected.</p> <p>RESERVED100 - Reserved (100) Operation Mode is selected.</p> <p>RESERVED101 - Reserved (101) Operation Mode is selected.</p>

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:OMODE

Description This command sets the K2 Operation mode for Linear and Ring Switching mode.

At *RST, this value is Reserved (000).

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:OMODE<wsp>RESERVED000|
RESERVED001|RESERVED010|RESERVED011|
UNI100|BID101|MSRDI110|MSAIS111|IDLE000
|BRIDGED001|BSWITCHED010|
ETPROTECTION011|RESERVED100|
RESERVED101

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:OMODE

Parameter(s)	<p>Operation Mode:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: RESERVED000 RESERVED001 RESERVED010 RESERVED011 UNI100 BID101 MSRDI110 MSAIS111 IDLE000 BRIDGED001 BSWITCHED010 ETPROTECTION011 RESERVED100 RESERVED101.</p> <p>Selects the K2 Operation Mode for Linear and Ring Switching.</p> <p>RESERVED000, Selects the Reserved (000) Operation Mode.</p> <p>RESERVED001, Selects the Reserved (001) Operation Mode.</p> <p>RESERVED010, Selects the Reserved (010) Operation Mode.</p> <p>RESERVED011, Selects the Reserved (011) Operation Mode.</p> <p>UNI100, selects the Unidirectional (100) Operatin Mode.</p> <p>BID101, selects the Bidirectional (101) Operation Mode.</p>
---------------------	--

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:OMODE

Parameter(s)	
	MSRDI110, selects the MS-RDI (110) Operation Mode.
	MSAIS111, Selects the MS-AIS (111) Operation Mode.
	IDLE000,Selects the IDLE (000) Operation Mode.
	BRIDGED001,Selects the BRIDGED (001) Operation Mode.
	BSWITCHED010,Selects the BSWITCHED (010) Operation Mode.
	ETPROTECTION011,Selects the (011) Operation Mode.
	RESERVED100,Selects the RESERVED (100) Operation Mode.
	RESERVED101,Selects the RESERVED (101) Operation Mode.
	RESERVED011 - Selects the Reserved (011) Operation Mode.
	UNI100 - Selects the Unidirectional (100) Operation Mode.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:OMODE?**

Description	<p>This query returns the K2 Operation mode for Linear and Ring Switching mode.</p> <p>At *RST, this value is Reserved (000).</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:OMODE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Type></p>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character Response Data> element.</p> <p>Returns the K2 Operation mode for Linear and Ring Switching mode.</p>

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:DNODE

Description This command selects K1 Destination node Id for Ring Switching mode.

At *RST, this value is 0.

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:DNODE <wsp> <Node Id>

Parameter(s) Node Id:
The program data syntax for the parameter is defined as a <Numeric Value> element.
Selects the K1 Destination Node Id for Ring Switching.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:DNODE?**

Description	<p>This query returns K1 Destination node Id for Ring Switching mode.</p> <p>At *RST, this value is 0.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:DNODE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Value></p>
Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element.</p> <p>Returns the K1 Destination Node Id for Ring Switching mode.</p>

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:SNODE

Description	<p>This command selects the K2 Source Node Id for Ring Switching mode.</p> <p>At *RST, this value is 0.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:SNODE <wsp> <Node Id></p>
Parameter(s)	<p>Node Id:</p> <p>The program data syntax for the parameter is defined as a <Numeric Value> element.</p> <p>Selects the K2 Source Node Id for Ring Switching.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:SNODE?**

Description	<p>This query returns K2 Source Node Id for Ring Switching mode.</p> <p>At *RST, this value is 0.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:SNODE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Value></p>
Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element.</p> <p>Returns the K2 Source Node Id for Ring Switching mode.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:BREQuest**

Description	<p>This command sets the K2 Bridge Request for Ring Switching mode.</p> <p>At *RST, this value is Short Path.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:BREQuest <wsp> SPATH LPATH</p>
Parameter(s)	<p>Bridge Request:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: SPATH LPATH.</p> <p>Selects the K2 Bridge Request for Ring Switching.</p> <p>SPATH - Selects Short Path Bridge Request.</p> <p>LPATH - Selects Long Path Bridge Request.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:BREQuest?**

Description	<p>This query returns the K2 Bridge Request for Ring Switching mode.</p> <p>At *RST, this value is Short Path.</p>
Syntax	<code>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:BREQuest?</code>
Parameter(s)	None
Response Syntax	<Type>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character Response Data> 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>

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:H:SSBits

Description	<p>This command selects SS Bits.</p> <p>At *RST, this value is 10.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:H:SSBits <wsp>00 01 10 11</p>
Parameter(s)	<p>SS Bits:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: 00 01 10 11.</p> <p>Selects H1 SS Bits.</p> <p>00, Selects 00 (Sonet) SS Bits.</p> <p>01, Selects 01 (Undefined) SS Bits.</p> <p>10, Selects 10 (SDH) SS Bits.</p> <p>11, Selects 11 (Under Defined) SS Bits.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:H:SSBits?**

Description	This query returns H1 SS Bits. At *RST, this value is 10.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:H:SSBits?
Parameter(s)	None
Response Syntax	<Type>
Response(s)	Type: The response data syntax for <Type> is defined as a <Character Response Data> element. Returns the SS Bits. 00, selects 00 (Sonet) SS Bits. 01, selects 01 (Undefined) SS Bits. 10, selects 10 (SDH) SS Bits. 11, selects 11 (Under Defined) SS Bits.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:H:OVERwrite**

Description This command enables or disables the overwrite for H1.

At *RST, this value is OFF.

Syntax :SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:H:OVERwrite <wsp> <Set>

Parameter(s) Set:
The program data syntax for the parameter is defined as a <Boolean Program Data> element.
Enables or disables Overwrite.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:H:OVERwrite?**

Description	This query returns the H1-Overwrite status. At *RST, this value is OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:H:OVERwrite?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 Numeric Response Data> element. Returns the Overwrite status.

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:OVERwrite

Description	<p>This command enables or disables the S1 Overwrite.</p> <p>At *RST, this value is OFF.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:OVERwrite <wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Enables or disables the S1 Overwrite.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:S:OVERwrite?**

Description	This query returns the S1-Overwrite status. At *RST, this value is OFF.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:OVERwrite?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 Numeric Response Data> element. Returns the Overwrite status.

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:S:BITS:VALue**

Description	<p>This command sets the S1-Bits 1-4 value.</p> <p>At *RST, this value is 0000.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:VALue <wsp> <S1 byte ></p>
Parameter(s)	<p>S1 byte:</p> <p>The program data syntax for the parameter is defined as a <Non Decimal Numeric Program Data> element.</p> <p>Sets the S1-Bits 1-4 value.</p>

**:SOURce[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:S:BITS:VALue?**

Description	This query returns the S1-Bits 1-4 value. At *RST, this value is 0000.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:VALue?
Parameter(s)	None
Response Syntax	<Value>
Response(s)	Value: The response data syntax for <Value> is defined as a <Binary Response Data> element. Returns the S1 Bits 1-4 value.

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:SSMessage

Description

This command selects the Bits 5- 8 (Synchronization Status Message) of the S1 byte to convey the synchronization status of the NE.

*RST, this value is Quality Unknown (0000).

Syntax

```
:SOURce[1..n]:DATA:TELEcom:SDHSonet:  
ADVanced:S:BITS:SSMessage <wsp>  
QUNKNOWN0000 | RESERVED0001 | ITUTG811 |  
RESERVED0011 | SSUA0100 | RESERVED0101 |  
RESERVED0110 | RESERVED0111 | SSUB1000 |  
RESERVED1001 | RESERVED1010 | ITUTG813 |  
RESERVED1100 | RESERVED1101 | RESERVED1110  
| DUSYNCH1111
```

:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:SSMessage

Parameter(s)

Synchronization Status Message:

The program data syntax for the parameter is defined as a <Character Program Data> element.

The allowed <Character Program Data> elements for this parameter are:

QUNKNOWN0000|RESERVED0001|ITUTG811|
RESERVED0011|SSUA0100|RESERVED0101|
RESERVED0110|RESERVED0111|SSUB1000|
RESERVED1001|RESERVED1010|ITUTG813|
RESERVED1100|RESERVED1101|RESERVED1111|
DUSYNCH1111.

Sets the Bits 5-8 (Synchronization Status Message).

QUNKNOWN0000, Selects Quality Unknown (0000) Bits Synchronization Status Message.

RESERVED0001, Selects Reserved (0001) Bits Synchronization Status Message.

ITUTG811, Selects ITU-T G.811 (0010) Bits Synchronization Status Message.

RESERVED0011, Selects Reserved (0011) Bits Synchronization Status Message.

SSUA0100, Selects SSU-A (0100) Bits Synchronization Status Message.

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

Parameter(s)	
	RESERVED0101, Selects Reserved (0101) Bits Synchronization Status Message.
	RESERVED0110, Selects Reserved (0110) Bits Synchronization Status Message.
	RESERVED0111, Selects Reserved (0111) Bits Synchronization Status Message.
	SSUB1000, Selects SSU-B (1000) Bits Synchronization Status Message.
	RESERVED1001, Selects Reserved (1001) Bits Synchronization Status Message.
	RESERVED1010, Selects Reserved (1010) Bits Synchronization Status Message.
	ITUTG813, Selects ITU-T G.813 Option I (SEC) (1011) Bits Synchronization Status Message.
	RESERVED1100, Selects Reserved (1100) Bits Synchronization Status Message.
	RESERVED1101, Selects Reserved (1101) Bits Synchronization Status Message.
	RESERVED1110, Selects Reserved (1110) Bits Synchronization Status Message.
	DUSYNCH1111, Selects Don't Use for Synchronization (1111) Bits Synchronization Status Message.

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

Description	<p>This query returns the the Bits 5- 8 (Synchronization Status Message) of the S1 byte to convey the synchronization status of the NE.</p> <p>*RST, this value is Quality Unknown (0000).</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:SSMessage?</p>
Parameter(s)	<p>None</p>

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

Response Syntax <Type>

Response(s)

Type:

The response data syntax for <Type> is defined as a <Character Response Data> element.

Returns the Bits 5 through 8 (Synchronization Status Message) of the S1 byte.

QUNKNOWN0000 - Quality Unknown (0000) Bits Synchronization Status Message is selected.

RESERVED0001 - Reserved (0001) Bits Synchronization Status Message is selected.

ITUTG811 - ITU-T G.811 (0010) Bits Synchronization Status Message is selected.

RESERVED0011 - Reserved (0011) Bits Synchronization Status Message is selected.

SSUA0100 - SSU-A (0100) Bits Synchronization Status Message is selected.

RESERVED0101 - Reserved (0101) Bits Synchronization Status Message is selected.

RESERVED0110 - Reserved (0110) Bits Synchronization Status Message is selected.

RESERVED0111 - Reserved (0111) Bits Synchronization Status Message is selected.

SSUB1000 - SSU-B (1000) Bits Synchronization Status Message is selected.

RESERVED1001 - Reserved (1001) Bits Synchronization Status Message is selected.

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

Response(s)	RESERVED1010- Reserved (1010) Bits Synchronization Status Message is selected.
	ITUTG813- ITU-T G.813 Option I (SEC) (1011) Bits Synchronization Status Message is selected.
	RESERVED1100- Reserved (1100) Bits Synchronization Status Message is selected.
	RESERVED1101- Reserved (1101) Bits Synchronization Status Message is selected.
	RESERVED1110- Reserved (1110) Bits Synchronization Status Message is selected.
	DUSYNCH1111- Don't Use for Synchronization (1111) Bits Synchronization Status Message is selected.

:SENSe[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:SMODE

Description	<p>This command selects the Switching mode for RX tabs.</p> <p>At *RST, this value is Linear.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:SMODE<wsp>LINEar RING</pre>
Parameter(s)	<p>Mode:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: LINEar RING.</p> <p>Selects the Switching mode for TX and RX tabs.</p> <p>LINEar, Selects the Linear switching mode.</p> <p>RING, Selects the Ring Switching mode.</p>

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:SMODE?**

Description	<p>This query returns the Switching mode for RX tabs.</p> <p>At *RST, this value is Linear.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:SMODE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Type></p>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character Response Data> element.</p> <p>Returns the Switching mode for TX and RX tabs.</p> <p>LINear - Linear is selected for Switching Mode.</p> <p>RING - Ring is selected for Switching Mode.</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:REQuest?**

Description	This query returns the K1 Request for Linear Switching mode for RX. At *RST, this value is --.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:REQuest?
Parameter(s)	None
Response Syntax	<Type>
Response(s)	Type: The response data syntax for <Type> is defined as a <Character Response Data> element. Returns the Request for Linear Switching mode.

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:REQuest?**

Description	<p>This query returns the K1 Request for Ring Switching mode for RX.</p> <p>At *RST, this value is --.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:RING:REQuest?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Type></p>
Response(s)	<p>Type: The response data syntax for <Type> is defined as a <Character Response Data> element. Returns the Request for Ring Switching mode.</p>

:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:CHANnel?

Description	<p>This query returns Channel for Linear Switching mode RX.</p> <p>At *RST, this value is --.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:CHANnel?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Value></p>
Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element.</p> <p>Returns the Channel for Linear Switching Mode.</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:PCHannel?**

Description	<p>This query returns Protected Channel for Linear Switching mode RX.</p> <p>At *RST, this value is --.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:PCHannel?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Value></p>
Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element.</p> <p>Returns the Protected Channel for Linear Switching mode.</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:ARCHitecture?**

Description	<p>This query returns Architecture for Linear Switching mode RX.</p> <p>At *RST, this value is --.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:ARCHitecture?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Type></p>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character Response Data> element.</p> <p>Returns the Architecture for Linear Switching mode.</p> <p>1TO1, 1+1 Architecture is selected.</p> <p>1TON, 1:N Architecture is selected.</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:OMODE?****Description**

This query returns Operation mode for Linear Switching mode RX.

At *RST, this value is --.

Syntax

:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:LINEar:OMODE?

:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:LINEar:OMODE?

Parameter(s) None

Response Syntax <Type>

Response(s) Type:
The response data syntax for <Type> is defined as a <Character Response Data> element.
Returns the Operation mode for Linear Switching mode.
RESERVED000 - Reserved (000) Operation Mode is selected.
RESERVED001 - Reserved (001) Operation Mode is selected.
RESERVED010 - Reserved (010) Operation Mode is selected.
RESERVED011 - Reserved (011) Operation Mode is selected.
UNI100 - Unidirectional (100) Operatin Mode is selected.
BID101 - Bidirectional (101) Operation Mode is selected.
MSRDI110 - MS-RDI (110) Operation Mode is selected.
MSAIS111 - MS-AIS (111) Operation Mode is selected.

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:OMODE?****Description**

This query returns Operation mode for Ring Switching mode RX.

At *RST, this value is --.

Syntax

:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:OMODE?

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:RING:OMODE?**

Parameter(s) None

Response Syntax <Type>

Response(s) Type:
The response data syntax for <Type> is defined as a <Character Response Data> element.
Returns the Operation mode for Ring Switching mode.
MSRDI110 - MS-RDI (110) Operation Mode is selected.
MSAIS111 - MS-AIS (111) Operation Mode is selected.
IDLE000 - Idle (000) Operation Mode is selected.
BRIDGED001 - Bridge (001) Operation Mode is selected.
BSWITCHED010 - Bridged and Switched (010) Operation Mode is selected.
ETPROTECTION011 - Extra Traffic - Protection (011) Operation Mode is selected.
RESERVED100 - Reserved (100) Operation Mode is selected.
RESERVED101 - Reserved (101) Operation Mode is selected.

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:DNODE?**

Description	<p>This query returns K1 Destination node Id for Ring Switching mode RX.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:DNODE?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Value></p>
Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element.</p> <p>Returns K1 Destination node Id for Ring Switching mode RX.</p>

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:SNODE?**

Description	This query returns K2 Source Node Id for Ring Switching mode RX. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:SNODE?
Parameter(s)	None
Response Syntax	<Value>
Response(s)	Value: The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element. Returns K2 Source Node Id for Ring Switching mode RX.

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:APS:K:BREQuest?**

Description	<p>This query returns K2 Bridge Request for Ring Switching mode RX.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:APS:K:BREQuest?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Type></p>
Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <Character Response Data> 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>

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:CAPTURE:TRANSitions**

Description This command allows to select the number of K1/K2 byte transitions that will be captured.

At *RST, this value is 50.

Syntax :SENSe[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:CAPTURE:TRANSitions <wsp>
<Number>

Parameter(s) Number:
The program data syntax for the parameter is defined as a <Numeric Value> element.
Allows to select the number of K1/K2 byte transitions that will be captured.

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:CAPTure:TRANSitions?**

Description	<p>This query returns the selected number of K1/K2 byte transitions that will be captured.</p> <p>At *RST, this value is 50.</p>
Syntax	<p>:SENSe[1..n]:DATA:TELEcom:SDHSonet: ADVanced:CAPTure:TRANSitions?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Value></p>
Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element.</p> <p>Allows to select the number of K1/K2 byte transitions that will be captured.</p>

**:SENSe[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:CAPTure**

Description This query allows to enable the K1/K2 capture process.

At *RST, this value is OFF.

Syntax :SENSe[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:CAPTure

Parameter(s) None

**:FETCh[1..n]:DATA:TELEcom:SDHSonet:
ADVanced:H:SSBits?**

Description	This query returns the S1 Bits 1-4 value RX. At *RST, this value is device dependent.
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:H:SSBits?
Parameter(s)	None
Response Syntax	<Value>
Response(s)	Value: The response data syntax for <Value> is defined as a <NR3 Numeric Response Data> element. Returns the Bits 5 and 6 of the H1 byte represent the ss bits.

:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:VALue?

Description	<p>This query returns S1 Bits 1-4 value of the RX.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:VALue?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Value></p>
Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <Binary Response Data> element.</p> <p>Returns the Bits 5 and 6 of the H1 byte represent the ss bits.</p>

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

Description	This query returns the Bits 5-8 (Synchronization Status Message) of the S1 byte. At *RST, this value is Quality Unknown (0000).
Syntax	:FETCh[1..n]:DATA:TELEcom:SDHSonet: ADVanced:S:BITS:SSMessage?
Parameter(s)	None
Response Syntax	<Type>

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

Response(s)

Type:

The response data syntax for <Type> is defined as a <Character Response Data> element.

Returns the Bits 5 through 8 (Synchronization Status Message) of the S1 byte.

QUNKNOWN0000 - Quality Unknown (0000) Bits Synchronization Status Message is selected.

RESERVED0001 - Reserved (0001) Bits Synchronization Status Message is selected.

ITUTG811 - ITU-T G.811 (0010) Bits Synchronization Status Message is selected.

RESERVED0011 - Reserved (0011) Bits Synchronization Status Message is selected.

SSUA0100 - SSU-A (0100) Bits Synchronization Status Message is selected.

RESERVED0101 - Reserved (0101) Bits Synchronization Status Message is selected.

RESERVED0110 - Reserved (0110) Bits Synchronization Status Message is selected.

RESERVED0111 - Reserved (0111) Bits Synchronization Status Message is selected.

SSUB1000 - SSU-B (1000) Bits Synchronization Status Message is selected.

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

Response(s)	RESERVED1001- Reserved (1001) Bits Synchronization Status Message is selected.
	RESERVED1010- Reserved (1010) Bits Synchronization Status Message is selected.
	ITUTG813- ITU-T G.813 Option I (SEC) (1011) Bits Synchronization Status Message is selected.
	RESERVED1100- Reserved (1100) Bits Synchronization Status Message is selected.
	RESERVED1101- Reserved (1101) Bits Synchronization Status Message is selected.
	RESERVED1110- Reserved (1110) Bits Synchronization Status Message is selected.
	DUSYNCH1111- Don't Use for Synchronization (1111) Bits Synchronization Status Message is selected.

Test Preferences Commands

:OUTPut[1..n]:TELEcom:SDHSonet:PREFerence :DPReferences:LASer:STATus

Description	<p>This command sets the Laser Control status of the default Test Preferences.</p> <p>At *RST, this value is set to ON.</p>
Syntax	<p>:OUTPut[1..n]:TELEcom:SDHSonet:PREFerences :DPReferences:LASer:STATus<wsp> <Set></p>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Sets the Laser Control status.</p>
Example(s)	<p>OUTPut:TELEcom:SDHSonet:PREFerences: DPReferences:LASER:STATus 1</p>
See Also	<p>OUTPut[1..n]:TELEcom:PREFerence:FIBer:LASer</p>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFErences:DPReferences:LASer:STATus?**

Description	<p>This query returns the Laser Control status of the default Test Preferences.</p> <p>At *RST, this value is set to ON.</p>
Syntax	:OUTPut[1..n]:TELEcom:SDHSonet:PREFErences :DPReferences:LASer:STATus?
Parameter(s)	None
Response Syntax	<Set>
Response(s)	<p>Set:</p> <p>The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the Laser Control status.</p>
Example(s)	<p>OUTPut:TELEcom:SDHSonet:PREFErences: DPReferences:LASER:STATus 1</p> <p>OUTPut:TELEcom:SDHSonet:PREFErences: DPReferences:LASER:STATus Returns 1</p>
See Also	OUTPut[1..n]:TELEcom:PREFErence:FIBer: LASer?

:OUTPut[1..n]:TELEcom:SDHSonet:PREFereNCes: :DPReferences:BFOVerwrite:STATus

Description	<p>This command sets the status of Bulk Filled Override. When the Bulk Filled Override check box is enabled, it fills up the bytes of the STS-1 SPE's columns 30 and 59 with the selected pattern from the tab Pattern TX.</p> <p>At *RST, this is enabled.</p>
Syntax	<pre>:OUTPut[1..n]:TELEcom:SDHSonet:PREFereNCes: :DPReferences:BFOVerwrite:STATus<wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>When the Bulk Filled Override check box is selected, it fills up the bytes of the STS-1 SPE's columns 30 and 59 with the selected pattern from the tab Pattern TX.</p>
Example(s)	<pre>OUTPut:TELEcom:SDHSonet:PREFereNCes: DPReferences:BFOVErwrite:STATus 1</pre>
See Also	<pre>OUTPut[1..n]:TELEcom:SDHSonet:PREFereNCes: DPReferences:SMLTest:RXLTraffic:STATus</pre>

**:OUTPut[1..n]:TELEcom:SDHSonet:PREFere
s:DPReferences:BFOVerwrite:STATus?**

Description	This query returns the status of Bulk Filled Override. At *RST, this is enabled.
Syntax	:OUTPut[1..n]:TELEcom:SDHSonet:PREFere s:DPReferences:BFOVerwrite:STATus?
Parameter(s)	None
Response Syntax	<Set>

:OUTPut[1..n]:TELEcom:SDHSonet:PREFerence s:DPReferences:BFOVERwrite:STATus?

Response**Set:**

The response data syntax for <Set> is defined as a <NRI NUMERIC RESPONSE DATA> element.

Returns the status of Bulk Filled Override. When the Bulk Filled Override check box is enabled, it fills up the bytes of the STS-1 SPE's columns 30 and 59 with the selected pattern from the tab Pattern TX.

Example(s)

OUTPut:TELEcom:SDHSonet:PREFerences:
DPReferences:BFOVERwrite:STATus 1

OUTPut:TELEcom:SDHSonet:PREFerences:
DPReferences:BFOVERwrite:STATus? Returns 1

See Also

OUTPut[1..n]:TELEcom:SDHSonet:PREFerences:
DPReferences:SMLTest:RXLTraffic:STATus?

**:OUTPut[1..n]:TELEcom:SDHSonet:PREFere
s:DPReferences:SMLTest:PATtern:TYPE****Description**

This command sets the type of configuration of the default TX/RX Test Pattern that will be used when starting a test case using SmartMode.

At *RST, this is set to PRBS 2²³-1.

Syntax

```
:OUTPut[1..n]:TELEcom:SDHSonet:PREFere  
s:DPReferences:SMLTest:PATtern:TYPE  
<wsp>PRBS2E9|PRBS2E11|PRBS2E15|  
PRBS2E20|PRBS2E23|PRBS2E31|1100|1010|  
1111|0000|1in8|1in16
```

**:OUTPut[1..n]:TELEcom:SDHSonet:PREFerence
s:DPreferences:SMLTest:PATtern:TYPE**

Parameter(s)	Type:
	<p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: PRBS2E9 PRBS2E11 PRBS2E15 PRBS2E20 PRBS2E23 PRBS2E31 1100 1010 1111 0000 1in8 1in16.</p> <p>Sets the type of configuration of the default TX/RX Test Pattern that will be used when starting a test case using SmartMode.</p> <p>PRBS2E9, selects PRBS2E9 as Test Pattern type. PRBS2E11, selects PRBS2E11 as Test Pattern type. PRBS2E15, selects PRBS2E15 as Test Pattern type. PRBS2E20, selects PRBS2E20 as Test Pattern type. PRBS2E23, selects PRBS2E23 as Test Pattern type. PRBS2E31, selects PRBS2E31 as Test Pattern type. 1100, selects 1100 as Test Pattern type. 1010, selects 1010 as Test Pattern type. 1111, selects 1111 as Test Pattern type. 0000, selects 0000 as Test Pattern type. 1in8, selects 1in8 as Test Pattern type. 1in16, selects 1in16 as Test Pattern type.</p>

**:OUTPut[1..n]:TELEcom:SDHSonet:PREFerence
s:DPreferences:SMLTest:PATtern:TYPE**

Example(s) OUTPut:TELEcom:SDHSonet:PREferences:
DPreferences:SMLTest:PATtern:TYPE 1in15

See Also SOURce[1..n]:DATA:TELEcom:PATtern:TYPE

:OUTPut[1..n]:TELEcom:SDHSonet: PREferences:DPreferences:SMLTest:PATtern: TYPE?

Description This query returns the type of configuration of the default TX/RX Test Pattern that will be used when starting a test case using SmartMode.

At *RST, this is set to PRBS 2²³-1.

Syntax :OUTPut[1..n]:TELEcom:SDHSonet:PREferences
:DPreferences:SMLTest:PATtern:TYPE?

Parameter(s) None

Response Syntax <Type>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPReferences:SMLTest:PATtern:
TYPE?****Response(s)**

Type:

The response data syntax for <Type> is defined as a <Character response data> element.

Returns the type of configuration of the default TX/RX Test Pattern that will be used when starting a test case using SmartMode.

PRBS2E9, PRBS2E9 is selected as Test Pattern type.

PRBS2E11, PRBS2E11 is selected as Test Pattern type.

PRBS2E15, PRBS2E15 is selected as Test Pattern type.

PRBS2E20, PRBS2E20 is selected as Test Pattern type.

PRBS2E23, PRBS2E23 is selected as Test Pattern type.

PRBS2E31, PRBS2E31 is selected as Test Pattern type.

1100, 1100 is selected as Test Pattern type.

1010, 1010 is selected as Test Pattern type.

1111, 1111 is selected as Test Pattern type.

0000, 0000 is selected as Test Pattern type.

1in8, 1in8 is selected as Test Pattern type.

1in16, 1in16 is selected as Test Pattern type.

:OUTPut[1..n]:TELEcom:SDHSonet: PREferences:DPReferences:SMLTest:PATtern: TYPE?

Example(s) OUTPut:TELEcom:SDHSonet:PREferences:
 DPReferences:SMLTest:PATtern:TYPE 1in16

 OUTPut:TELEcom:SDHSonet:PREferences:
 DPReferences:SMLTest:PATtern:TYPE? Returns
 1in16

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

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPreferences:SMLTest:INVert:
STATus**

Description	<p>This command allows the inversion of the test pattern. When enabled, every 0 in the pattern will be changed for 1 and every 1 for 0.</p> <p>At *RST, this is set to disabled.</p>
Syntax	<pre>:OUTPut[1..n]:TELEcom:SDHSonet:PREferences :DPreferences:SMLTest:INVert:STATus <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Allows the inversion of the test pattern. When enabled, every 0 in the pattern will be changed for 1 and every 1 for 0.</p>
Example(s)	<pre>OUTPut:TELEcom:SDHSonet:PREferences: DPreferences:SMLTest:INVert:STATus 1</pre>
See Also	<pre>OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPreferences:LASER:STATus</pre>

:OUTPut[1..n]:TELEcom:SDHSonet: PREferences:DPreferences:SMLTest:INVert: STATus?

Description	<p>This query returns the inversion of the test pattern. When enabled, every 0 in the pattern will be changed for 1 and every 1 for 0.</p> <p>At *RST, this is set to disabled.</p>
Syntax	<code>:OUTPut[1..n]:TELEcom:SDHSonet:PREferences :DPreferences:SMLTest:INVert:STATus?</code>
Parameter(s)	None
Response Syntax	<Set>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFerences:DPReferences:SMLTest:INVert:
STATus?****Response(s)**

Set:

The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the type of configuration of the default TX/RX Test Pattern that will be used when starting a test case using SmartMode.

PRBS2E9, PRBS2E9 is selected as Test Pattern type.

PRBS2E11, PRBS2E11 is selected as Test Pattern type.

PRBS2E15, PRBS2E15 is selected as Test Pattern type.

PRBS2E20, PRBS2E20 is selected as Test Pattern type.

PRBS2E23, PRBS2E23 is selected as Test Pattern type.

PRBS2E31, PRBS2E31 is selected as Test Pattern type.

1100, 1100 is selected as Test Pattern type.

1010, 1010 is selected as Test Pattern type.

1111, 1111 is selected as Test Pattern type.

0000, 0000 is selected as Test Pattern type.

1in8, 1in8 is selected as Test Pattern type.

1in16, 1in16 is selected as Test Pattern type.

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFerences:DPreferences:SMLTest:INVert:
STATus?**

Example(s) OUTPut:TELEcom:SDHSonet:PREFerences:
 DPreferences:SMLTest:INVert:STATus 1

 OUTPut:TELEcom:SDHSonet:PREFerences:
 DPreferences:SMLTest:INVert:STATus? Returns 1

See Also OUTPut[1..n]:TELEcom:SDHSonet:PREFerences:
 DPreferences:LASER:STATus?

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPReferences:SMLTest:
RXLTraffic:STATUs**

Description	<p>This command analyzes the line traffic without test pattern thus squelching the pattern loss and bit error indication.</p> <p>At *RST, this is set to disabled.</p>
Syntax	<pre>:OUTPut[1..n]:TELEcom:SDHSonet:PREferences :DPReferences:SMLTest:RXLTraffic:STATUs <wsp> <Set></pre>
Parameter(s)	<p>Set:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Analyzes the line traffic without test pattern thus squelching the pattern loss and bit error indication.</p>
Example(s)	<pre>OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:SMLTest:RXLTraffic:STATUs 1</pre>
See Also	<pre>OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPReferences:BFOVwrite:STATUs</pre>

:OUTPut[1..n]:TELEcom:SDHSonet: PREferences:DPreferences:SMLTest: RXLTraffic:STATus?

Description	This query returns the status of line traffic without test pattern thus squelching the pattern loss and bit error indication. At *RST, this is set to disabled.
Syntax	:OUTPut[1..n]:TELEcom:SDHSonet:PREferences :DPreferences:SMLTest:RXLTraffic:STATus?
Parameter(s)	None
Response Syntax	<Set>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPReferences:SMLTest:
RXLTraffic:STATus?**

Response(s)	Set: The response data syntax for <Set> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the line traffic without test pattern thus squelching the pattern loss and bit error indication.
Example(s)	OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:SMLTest:RXLTraffic:STATus 1 OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:SMLTest:RXLTraffic:STATus? Returns 1
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPReferences:BFOVerwrite:STATus?

:OUTPut[1..n]:TELEcom:SDHSonet: PREFerences:DPreferences:BACKground: OTNFlex

Description

This command allows the selection of the ODU FLEX multiplexed background traffic.

At *RST, this value is set to AIS.

Syntax

```
:OUTPut[1..n]:TELEcom:SDHSonet:PREFerences  
:DPreferences:BACKground:OTNFlex <wsp> AIS  
|NCLient|PRBS31|UNALlocated
```

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFErences:DPreferences:BACKground:
OTNFlex**

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: AIS NCLient PRBS31 UNALLocated.</p> <p>Allows the selection of the ODU FLEX multiplexed background traffic.</p> <p>AIS, selects AIS as the ODU FLEX multiplexed background traffic.</p> <p>NCLient, selects NCLient as the ODU FLEX multiplexed background traffic.</p> <p>PRBS31, selects PRBS31 as the ODU FLEX multiplexed background traffic.</p> <p>UNALLocated, selects UNALLocated as the ODU FLEX multiplexed background traffic.</p>
Example(s)	<p>OUTPut:TELEcom:SDHSonet:PREFErences: DPreferences:BACKground:OTNFlex PRBS31</p>
See Also	<p>OUTPut[1..n]:TELEcom:SDHSonet:PREFErences: DPreferences:SMLTest:PATtern:TYPE</p>

:OUTPut[1..n]:TELEcom:SDHSonet: PREFerences:DPReferences:BACKground: OTNFlex?

Description	This query returns the selected ODU FLEX multiplexed background traffic. At *RST, this value is set to AIS.
Syntax	:OUTPut[1..n]:TELEcom:SDHSonet:PREFerences :DPReferences:BACKground:OTNFlex?
Parameter(s)	None
Response Syntax	<Type>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPreferences:BACKground:
OTNFlex?**

Response(s)	<p>The response data syntax for <Type> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the selection of the ODU FLEX multiplexed background traffic.</p> <p>AIS, AIS is selected as the ODU FLEX multiplexed background traffic.</p> <p>NCLIENT, NCLIENT is selected as the ODU FLEX multiplexed background traffic.</p> <p>PRBS31, PRBS31 is selected as the ODU FLEX multiplexed background traffic.</p> <p>UNALLOCATED, UNALLOCATED is selected as the ODU FLEX multiplexed background traffic.</p>
Example(s)	<p>OUTPut:TELEcom:SDHSonet:PREferences: DPreferences:BACKground:OTNFlex PRBS31</p> <p>OUTPut:TELEcom:SDHSonet:PREferences: DPreferences:BACKground:OTNFlex? Returns PRBS31</p>
See Also	<p>OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPreferences:SMLTest:PATtern:TYPE?</p>

:OUTPut[1..n]:TELEcom:SDHSonet: PREFErences:DPReferences:BACKground:OTN

Description This command allows the selection of the ODU multiplexed background traffic.

At *RST, this value is set to PRBS31 pattern.

Syntax :OUTPut[1..n]:TELEcom:SDHSonet:PREFErences
:DPReferences:BACKground:OTN <wsp>
AIS|PRBS31|NCLient

:OUTPut[1..n]:TELEcom:SDHSonet: PREFErences:DPReferences:BACKground:OTN

Parameter(s)	<p><Type></p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: AIS PRBS31 NCLient.</p> <p>Allows the selection of the ODU multiplexed background traffic.</p> <p>AIS, selects AIS as the ODU multiplexed background traffic.</p> <p>PRBS31, selects PRBS31 pattern as the ODU multiplexed background traffic.</p> <p>NCLient, selects NULL Client (All Zeros) as the ODU multiplexed background traffic.</p>
Example(s)	<p>OUTPut:TELEcom:SDHSonet:PREFErences: DPReferences:BACKground:OTN PRBS31</p>
See Also	<p>OUTPut[1..n]:TELEcom:SDHSonet:PREFErences: DPReferences:BACKground:OTNFlex</p>

:OUTPut[1..n]:TELecom:SDHSonet: PREFErences:DPReferences:BACKground:OTN?

Description This query returns the selected ODU multiplexed background traffic.

At *RST, this value is set to PRBS31 pattern.

Syntax :OUTPut[1..n]:TELecom:SDHSonet:PREFErences
:DPReferences:BACKground:OTN?

Parameter(s) None

Response Syntax <Type>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFerences:DPReferences:BACKground:OTN?****Response(s)****Type:**

The response data syntax for <Type> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the selected ODU multiplexed background traffic.

AIS, AIS is selected as the ODU multiplexed background traffic.

PRBS31, PRBS31 pattern is selected as the ODU multiplexed background traffic.

NCLIENT, NULL Client (All Zeros) is selected as the ODU multiplexed background traffic.

Example(s)

OUTPut:TELEcom:SDHSonet:PREFerences:
DPReferences:BACKground:OTN PRBS31

OUTPut:TELEcom:SDHSonet:PREFerences:
DPReferences:BACKground:OTN? Returns
PRBS31

See Also

OUTPut[1..n]:TELEcom:SDHSonet:PREFerences:
DPReferences:BACKground:OTNFlex?

:OUTPut[1..n]:TELEcom:SDHSonet: PREFerences:DPReferences:BACKground:HOP

Description This command allows the selection of the default high order path background traffic.

At *RST, this value is set to Equipped.

Syntax :OUTPut[1..n]:TELEcom:SDHSonet:PREFerences
:DPReferences:BACKground:HOP <wsp>
AIS | EQUipped | UNEQuipped

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFErences:DPReferences:BACKground:HOP**

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: AIS EQUipped UNEQuipped.</p> <p>Allows the selection of the default high order path background traffic.</p> <p>AIS, selects AIS as the default high order path background traffic.</p> <p>EQUipped, selects EQUipped as the default high order path background traffic.</p> <p>UNEQuipped, selects UNEQuipped as the default high order path background traffic.</p>
Example(s)	OUTPut:TELEcom:SDHSonet:PREFErences: DPReferences:BACKground:HOP AIS
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREFErences: DPReferences:BACKground:OTN

:OUTPut[1..n]:TELEcom:SDHSonet: PREFerences:DPReferences:BACKground: HOP?

Description	This query returns the selected default high order path background traffic. At *RST, this value is set to Equipped.
Syntax	:OUTPut[1..n]:TELEcom:SDHSonet:PREFerences :DPReferences:BACKground:HOP?
Parameter(s)	None
Response Syntax	<Type>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPreferences:BACKground:
HOP?****Response(s)****Type:**

The response data syntax for <Type> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the selected default high order path background traffic.

AIS, AIS is selected as the default high order path background traffic.

EQUIPPED, EQUIPPED is selected as the default high order path background traffic.

UNEQUIPPED, UNEQUIPPED is selected as the default high order path background traffic.

Example(s)

OUTPut:TELEcom:SDHSonet:PREferences:
DPreferences:BACKground:HOP AIS

OUTPut:TELEcom:SDHSonet:PREferences:
DPreferences:BACKground:HOP? Returns AIS

See Also

OUTPut[1..n]:TELEcom:SDHSonet:PREferences:
DPreferences:BACKground:OTN?

:OUTPut[1..n]:TELEcom:SDHSonet: PREferences:DPreferences:BACKground:LOP

Description This command allows the selection of the default low order path background traffic.

At *RST, this value is set to Equipped.

Syntax :OUTPut[1..n]:TELEcom:SDHSonet:PREferences
:DPreferences:BACKground:LOP <wsp>
AIS | EQUipped | UNEQuipped

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPReferences:BACKground:LOP**

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: AIS EQUipped UNEQuipped.</p> <p>Allows the selection of the default low order path background traffic.</p> <p>AIS, selects AIS as the default low order path background traffic.</p> <p>EQUipped, selects EQUipped as the default low order path background traffic.</p> <p>UNEQuipped, selects UNEQuipped as the default low order path background traffic.</p>
Example(s)	OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:BACKground:LOP AIS
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPReferences:BACKground:HOP

:OUTPut[1..n]:TELEcom:SDHSonet:PREFerences:DPReferences:BACKground:LOP?

Description	This query returns the selection of the default low order path background traffic. At *RST, this value is set to Equipped.
Syntax	:OUTPut[1..n]:TELEcom:SDHSonet:PREFerences:DPReferences:BACKground:LOP?
Parameter(s)	None
Response Syntax	<Type>

**:OUTPut[1..n]:TELEcom:SDHSonet:PREFErences:
s:DPReferences:BACKground:LOP?**

Response(s)	<p>Type:</p> <p>The response data syntax for <Type> is defined as a <CHARACTER RESPONSE DATA> element.</p> <p>Returns the selected default low order path background traffic.</p> <p>AIS, AIS is selected as the default low order path background traffic.</p> <p>EQUIPPED, EQUIPPED is selected as the default low order path background traffic.</p> <p>UNEQUIPPED, UNEQUIPPED is selected as the default low order path background traffic.</p>
Example(s)	<p>OUTPut:TELEcom:SDHSonet:PREFErences: DPReferences:BACKground:LOP AIS</p> <p>OUTPut:TELEcom:SDHSonet:PREFErences: DPReferences:BACKground:LOP? Returns AIS</p>
See Also	<p>OUTPut[1..n]:TELEcom:SDHSonet:PREFErences: DPReferences:BACKground:HOP?</p>

:OUTPut[1..n]:TELEcom:SDHSonet: PREFerences:DPReferences:BACKground: DSNPdh

Description This command allows the selection of the default timeslot background traffic.

At *RST, this value is set to AIS.

Syntax :OUTPut[1..n]:TELEcom:SDHSonet:PREFerences
:DPReferences:BACKground:DSNPdh<wsp>AIS
|ALLZero

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFErences:DPreferences:BACKground:
DSNPdh**

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: AIS ALLZero.</p> <p>Allows the selection of the default time slot background traffic.</p> <p>AIS, selects AIS as the default time slot background traffic.</p> <p>ALLZero, selects All zeros as the default time slot background traffic.</p>
Example(s)	OUTPut:TELEcom:SDHSonet:PREFErences: DPreferences:BACKground:DSNPdh AIS
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREFErences: DPreferences:BACKground:LOP

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFErences:DPReferences:BACKground:
DSNPdh?**

Description	This query returns the selection of the default timeslot background traffic. At *RST, this value is set to AIS.
Syntax	:OUTPut[1..n]:TELEcom:SDHSonet:PREFErences :DPReferences:BACKground:DSNPdh?
Parameter(s)	None
Response Syntex	<Type>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFErences:DPreferences:BACKground:
DSNPdh?**

Response(s)

Type:

The response data syntax for <Type> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the selected default time slot background traffic.

AIS, AIS is selected as the default time slot background traffic.

ALLZERO, ALLZERO is selected as the default time slot background traffic.

Example(s)

OUTPut:TELEcom:SDHSonet:PREFErences:
DPreferences:BACKground:DSNPdh? Returns
AIS

See Also

OUTPut[1..n]:TELEcom:SDHSonet:PREFErences:
DPreferences:BACKground:LOP?

:OUTPut[1..n]:TELEcom:SDHSonet: PREFerences:DPreferences:LCAS:SOURce: STATus

Description

This command sets the status of LCAS Auto-Add at Startup. This setting allows to enable by default the Add Member(s) at Start for Source and Sink every time a test is created manually using Test Setup or when using SmartMode.

At *RST, this is set to enable.

Syntax

```
:OUTPut[1..n]:TELEcom:SDHSonet:PREFerences  
:DPreferences:LCAS:SOURce:STATus<wsp>  
<Set>
```

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFerences:DPreferences:LCAS:SOURce:
STATUs**

Parameter(s)	Set: The program data syntax for the parameter is defined as a <Boolean Program Data> element. Sets the status of LCAS Auto-Add at Startup. This setting allows to enable by default the Add Member(s) at Start for Source and Sink every time a test is created manually using Test Setup or when using SmartMode.
Example(s)	OUTPut:TELEcom:SDHSonet:PREFerences :DPreferences:LCAS:SOURce:STATUs 1
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREFerences: DPreferences:SMLTest:RXLTraffic:STATUs

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPreferences:LCAS:SOURce:
STATus?**

Description	This This query returns the status of LCAS Auto-Add at Startup. At *RST, this is set to enable.
Syntax	:OUTPut[1..n]:TELEcom:SDHSonet:PREferences :DPreferences:LCAS:SOURce:STATus?
Parameter(s)	None
Response Syntax	<Set>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFerences:DPreferences:LCAS:SOURce:
STATus?**

Response(s)	Set: The response data syntax for <Set> is defined as a <Character response data> element. Returns the status of LCAS Auto-Add at Startup.
Example(s)	OUTPut:TELEcom:SDHSonet:PREFerences: DPreferences:LCAS:SOURce:STATus 1 OUTPut:TELEcom:SDHSonet:PREFerences: DPreferences:LCAS:SOURce:STATus? Returns 1
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREFerences: DPreferences:SMLTest:RXLTraffic:STATus?

:OUTPut[1..n]:TELEcom:SDHSonet: PREFErences:DPReferences:OCSTm: COMPUtation: METHod

Description

This command allows to select the default method used to calculate the REI-L/MS-REI error for OC-192 and STM-64 interfaces.

At *RST, this value is set to M1 Only.

Syntax

:OUTPut[1..n]:TELEcom:SDHSonet:PREFErences
:DPReferences:OCSTm:COMPUtation:METHod
<wsp>M1ONLY|M0M1

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFerences:DPReferences:OCSTm:
COMPUtation: METHOd**

Parameter(s)	<p>Type:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>The allowed <Character Program Data> elements for this parameter are: M1ONLY M0M1.</p> <p>Allows to select the default method used to calculate the REI-L/MS-REI error for OC-192 and STM-64 interfaces.</p> <p>M1ONLY, selects e M1 Only as the default method used to calculate the REI-L/MS-REI error for OC-192 and STM-64 interfaces.</p> <p>M0M1, selects e M0 Only as the default method used to calculate the REI-L/MS-REI error for OC-192 and STM-64 interfaces.</p>
Example(s)	<pre>OUTPut:TELEcom:SDHSonet:PREFerences :DPReferences:OCSTm:COMPUtation:METHOd M0M1</pre>
See Also	<pre>OUTPut[1..n]:TELEcom:SDHSonet:PREFerences: DPReferences:BACKground:DSNPdh</pre>

:OUTPut[1..n]:TELEcom:SDHSonet: PREFErences:DPREFERences:OCSTm: COMPUtation:METhod?

Description	This query returns the selected default method used to calculate the REI-L/MS-REI error for OC-192 and STM-64 interfaces. At *RST, this value is set to M1 Only.
Syntax	:OUTPut[1..n]:TELEcom:SDHSonet:PREFErences :DPREFERences:OCSTm:COMPUtation:METhod?
Parameter(s)	None
Response Syntax	<Type>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFErences:DPReferences:OCSTm:
COMPUtation:METhod?****Response(s)**

Type:

The response data syntax for <Type> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the selected default method used to calculate the REI-L/MS-REI error for OC-192 and STM-64 interfaces.

M1ONLY, M1 ONLY is selected as the default method used to calculate the REI-L/MS-REI error for OC-192 and STM-64 interfaces.

M0M1, M0 ONLY is selected as the default method used to calculate the REI-L/MS-REI error for OC-192 and STM-64 interfaces.

Example(s)

OUTPut:TELEcom:SDHSonet:PREFErences:
DPReferences:OCSTm:COMPUtation:METhod
M0M1

OUTPut:TELEcom:SDHSonet:PREFErences:
DPReferences:OCSTm:COMPUtation:METhod?
Returns M0M1

See Also

OUTPut[1..n]:TELEcom:SDHSonet:PREFErences:
DPReferences:BACKground:DSNPdh?

:OUTPut[1..n]:TELEcom:SDHSonet: PREferences:DPreferences:DSN:LCODE: CONFiguration:NAME

Description

This command sets each loop code name.

At *RST, this value is set to Loop Code 1.

Syntax

:OUTPut[1..n]:TELEcom:SDHSonet:PREferences
:DPreferences:DSN:LCODE:CONFiguration:
NAME <wsp> <Row no> <Name>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPReferences:DSN:LCODE:
CONFIguration:NAME**

Parameter(s)	Row no: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the ten rows of loop code name. Name: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the loop code name. The name field allows up to 16 characters long.
Example(s)	OUTPut:TELEcom:SDHSonet:PREferences :DPReferences:DSN:LCODE:CONFIguration: NAME 1,Loop1
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFIguration:LUP

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPreferences:DSN:LCODE:
CONFIguration:NAME?**

Description	<p>This query returns the configured loop code name.</p> <p>At *RST, this value is set to Loop Code 1.</p>
Syntax	<p>:OUTPut[1..n]:TELEcom:SDHSonet:PREferences :DPreferences:DSN:LCODE:CONFIguration: NAME?<wsp><Row no> <Name></p>
Parameter(s)	<p>Row no: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the ten rows of loop code name.</p> <p>Name: The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element. Sets the loop code name. The name field allows up to 16 characters long.</p>
Response Syntax	<p><Name></p>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPReferences:DSN:LCODE:
CONFiguration:NAME?**

Response(s)	Name: The response data syntax for <Name> is defined as a <STRING RESPONSE DATA> element. Returns the configured loop code name.
Example(s)	OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFiguration: NAME 1,Loop1 OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFiguration: NAME? 1 Returns Loop1
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFiguration:LUP?

:OUTPut[1..n]:TELEcom:SDHSonet: PREferences:DPReferences:DSN:LCODE: CONFiguration:LUP

Description

This command sets the loop up values.

At *RST, this value is set to 10000.

Syntax

```
:OUTPut[1..n]:TELEcom:SDHSonet:PREferences  
:DPReferences:DSN:LCODE:CONFiguration:LUP  
<wsp> <Row no> <Value>
```

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPreferences:DSN:LCODE:
CONFiguration:LUP**

Parameter(s)	Row no: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the ten rows of Loop Up values. Value: The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element. Sets the Loop Up value from 000 to 1111111111111111.
Example(s)	OUTPut:TELEcom:SDHSonet:PREferences :DPreferences:DSN:LCODE:CONFiguration:LUP 1,001
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPreferences:DSN:LCODE:CONFiguration:NAME

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPreferences:DSN:LCODE:
CONFiguration:LUP?**

Description This query returns the configured loop up values.

At *RST, this value is set to 10000.

Syntax :OUTPut[1..n]:TELEcom:SDHSonet:PREferences
:DPreferences:DSN:LCODE:CONFiguration:LUP?
<wsp><Row no>

Parameter(s) Row no:
The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.
Sets the ten rows of Loop Up values.

Response Syntax <Value>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPReferences:DSN:LCODE:
CONFIguration:LUP?**

Response(s)	Value: The response data syntax for <Value> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the configured loop up values.
Example(s)	OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFIguration:LUP 1,001 OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFIguration:LUP? 1 Returns 001
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFIguration: NAME?

:OUTPut[1..n]:TELEcom:SDHSonet: PREFerences:DPReferences:DSN:LCODE: CONFIguration:LDOWN

Description

This command sets the configured loop down values.

At *RST, this value is set to 100.

Syntax

:OUTPut[1..n]:TELEcom:SDHSonet:PREFerences
:DPReferences:DSN:LCODE:CONFIguration:
LDOWN<wsp><Row no>,<Value>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREFerences:DPReferences:DSN:LCODE:
CONFIguration:LDOWN**

Parameter(s)	Row no: The program data syntax for the parameter is defined as a <Character Program Data> element. Sets the ten rows of Loop down values. Value: The program data syntax for the parameter is defined as a <NONDECIMAL NUMERIC PROGRAM DATA> element. Sets the Loop Down value from 000 to 1111111111111111.
Example(s)	OUTPut:TELEcom:SDHSonet:PREFerences: DPReferences:DSN:LCODE:CONFIguration: LDOWN 1,010
See Also	OUTPut[1..n]:TELEcom:SDHSonet:PREFerences: DPReferences:DSN:LCODE:CONFIguration:LUP

:OUTPut[1..n]:TELEcom:SDHSonet: PREferences:DPreferences:DSN:LCODE: CONFIguration:LDOWN?

Description	<p>This query returns the configured loop down values.</p> <p>At *RST, this value is set to 100.</p>
Syntax	<p>:OUTPut[1..n]:TELEcom:SDHSonet:PREferences :DPreferences:DSN:LCODE:CONFIguration: LDOWN? <wsp> <Row no></p>
Parameter(s)	<p>Row no:</p> <p>The program data syntax for the parameter is defined as a <Character Program Data> element.</p> <p>Sets the ten rows of Loop down values.</p>
Response Syntax	<p><Value></p>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPReferences:DSN:LCODE:
CONFiguration:LDOWN?**

Response(s)	<p>Value:</p> <p>The response data syntax for <Value> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the configured loop up values.</p>
Example(s)	<p>OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFiguration: LDOWn 1,010</p> <p>OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFiguration: LDOWn? 1 Returns 010</p>
See Also	<p>OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFiguration:LUP?</p>

:OUTPut[1..n]:TELEcom:SDHSonet: PREferences:DPReferences:DSN:LCODE: CONFIguration:EXPort

Description	<p>This command allows to save loop codes to a file.</p> <p>This command is not associated to any *RST value.</p>
Syntax	<pre>:OUTPut[1..n]:TELEcom:SDHSonet:PREferences :DPReferences:DSN:LCODE:CONFIguration: EXPort <wsp> <Path></pre>
Parameter(s)	<p>Path:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Save loop codes to a file.</p>
Example(s)	<pre>OUTPut:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFIguration: EXPort "C:\LoopCodes0.dlcf"</pre>
See Also	<pre>OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPReferences:DSN:LCODE:CONFIguration:l MPort</pre>

**:OUTPut[1..n]:TELEcom:SDHSonet:
PREferences:DPreferences:DSN:LCODE:
CONFiguration:IMPort**

Description	<p>This command allows to import loop codes from a previously saved file.</p> <p>This command is not associated to any *RST value.</p>
Syntax	<pre>:OUTPut[1..n]:TELEcom:SDHSonet:PREferences :DPreferences:DSN:LCODE:CONFiguration: IMPort <wsp> <Path></pre>
Parameter(s)	<p>Path:</p> <p>The program data syntax for the parameter is defined as a <STRING PROGRAM DATA> element.</p> <p>Import loop codes from a previously saved file.</p>
Example(s)	<pre>OUTPut:TELEcom:SDHSonet:PREferences: DPreferences:DSN:LCODE:CONFiguration: IMPort "C:\LoopCodes0.dlc"</pre>
See Also	<pre>OUTPut[1..n]:TELEcom:SDHSonet:PREferences: DPreferences:DSN:LCODE:CONFiguration: EXPort</pre>

Client Frequency Offset

:SOURce[1..n]:DATA:TELEcom:ETHernet: COFFset:FREQuency:OFFSet

Description This command allows entering a positive or a negative client frequency offset in ppm.

At *RST, this value is set to 0.

Syntax :SOURce[1..n]:DATA:TELEcom:ETHernet:
COFFset:FREQuency:OFFSet <wsp> <CFO>

Parameter(s) CFO:
The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.
Allows entering a positive or a negative client frequency offset in ppm.

Example(s) SOURce:DATA:TELEcom:ETHernet:COFFset:
FREQuency:OFFSet 55

See Also SOURce[1..n]:DATA:TELEcom:OTN:ERRor:OTU
[1..n]:F:AMOUNT

**SOURce[1..n]:DATA:TELEcom:ETHernet:
COFFset:FREQuency:OFFSet?**

Description	This query returns the entered positive or a negative client frequency offset in ppm. At *RST, this value is set to 0.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet: COFFset:FREQuency:OFFSet?
Parameter(s)	None
Response Syntax	<CFO>

**SOURce[1..n]:DATA:TELEcom:ETHernet:
COFFset:FREQUency:OFFSet?**

Response(s)	<p>CFO:</p> <p>The response data syntax for <CFO> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the entered positive or a negative client frequency offset in ppm.</p>
Example(s)	<p>SOURce:DATA:TELEcom:ETHernet:COFFset: FREQUency:OFFSet 55</p> <p>SOURce:DATA:TELEcom:ETHernet:COFFset: FREQUency:OFFSet? Returns 55</p>
See Also	<p>SOURce[1..n]:DATA:TELEcom:OTN:ERRor: OTU[1..n]:F:AMOUnt?</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset
:FREQuency?**

Description	This query returns the frequency used for transmission.
Syntax	:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset: FREQuency? <wsp>AFREquency NFREquency
Parameter(s)	<p>Frequency Type:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>The allowed <CHARACTER PROGRAM DATA> 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>
Response Syntax	<Frequency>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset
:FREQUENCY?**

Response(s)	<p>Frequency:</p> <p>The response data syntax for <Frequency> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the frequency used for transmission.</p> <p>AFREQUENCY, Actual Frequency indicates the frequency (Nominal frequency + port frequency offset + client frequency offset) used for transmission of the client signal.</p> <p>NFREQUENCY, Nominal Frequency indicates the nominal frequency of the signal.</p>
Example(s)	<p>FETCh:DATA:TELEcom:ETHernet:COFFset:FREQUENCY? NFREQUENCY Returns Frequency</p>
See Also	<p>SENSE[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQUENCY?</p>

**:SOURCE[1..n]:DATA:TELEcom:ETHernet:
COFFset:FREQuency:ENABLE**

Description	<p>This command allows to enable the frequency offset measurements.</p> <p>At *RST, this value is set to enable.</p>
Syntax	<pre>:SOURCE[1..n]:DATA:TELEcom:ETHernet: COFFset:FREQuency:ENABLE <wsp> <Status></pre>
Parameter(s)	<p>Status:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>Enables and disables the frequency offset measurements.</p>
Example(s)	<pre>SOURCE:DATA:TELEcom:ETHernet:COFFset:FRE Quency:ENABLE 1</pre>
See Also	<pre>SOURCE[1..n]:DATA:TELEcom:DS:ENABLEd</pre>

:SOURce[1..n]:DATA:TELEcom:ETHernet: COFFset:FREQuency:ENABle?

Description	This query returns the enable or disable status of the frequency offset measurements. At *RST, this value is set to enable.
Syntax	:SOURce[1..n]:DATA:TELEcom:ETHernet: COFFset:FREQuency:ENABle?
Parameter(s)	None
Response Syntax	<Status>

**:SOURce[1..n]:DATA:TELEcom:ETHernet:
COFFset:FREQuency:ENABLE?**

Response(s)	Status: The response data syntax for <Status> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of the frequency offset measurements.
Example(s)	SOURce:DATA:TELEcom:ETHernet:COFFset:FREQuency:ENABLE 1 SOURce:DATA:TELEcom:ETHernet:COFFset:FREQuency:ENABLE? Returns 1
See Also	SOURce[1..n]:DATA:TELEcom:DS:ENABLEd?

**:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset
:ALARm:HISTory?**

Description	<p>This query returns the history status of the Alarm analysis.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset: ALARm:HISTory? <wsp> Frequency </p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>Returns the history status of the Alarm analysis.</p> <p>Frequency alarm indicates that the received client signal rate meets the standard rate specifications (green) or not (red).</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset
:ALARm:HISTory?****Response Syntax** <History>**Response(s)**

History:

The response data syntax for <History> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the history status of the Alarm analysis.

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)

FETCh:DATA:TELEcom:ETHernet:COFFset:ALARm:HISTory? FREQuency Returns FREQuency History

**:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset
:ALARm:CURRent?**

Description	<p>This query returns the current status of the Alarm analysis.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset: ALARm:CURRent? <wsp> Frequency </p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>Returns the current status of the Alarm analysis.</p> <p>Frequency alarm indicates that the received client signal rate meets the standard rate specifications (green) or not (red).</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset
:ALARm:CURRent?****Response Syntax** <Current>**Response(s)**

Current:

The response data syntax for <Current> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current status of the Alarm analysis.

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)

FETCh:DATA:TELEcom:ETHernet:COFFset:ALARm: CURRent? FREQuency Returns FREQuency CURRent

**:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset:
:ALARm:SECOnds?**

Description	<p>This query returns the number of seconds for the alarm analysis.</p> <p>At *RST, this value is device dependent.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset: ALARm:SECOnds? Frequency </p>
Parameter(s)	<p>Alarm:</p> <p>The program data syntax for the parameter is defined as a <CHARACTER PROGRAM DATA> element.</p> <p>Returns the number of seconds for the Alarm analysis.</p> <p>Frequency alarm indicates that the received client signal rate meets the standard rate specifications (green) or not (red).</p>

**:FETCh[1..n]:DATA:TELEcom:ETHernet:COFFset
:ALARm:SEConds?****Response Syntax** <Seconds>**Response(s)**

Seconds:

The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the number of seconds for the alarm analysis.

Example(s)

FETCh:DATA:TELEcom:ETHernet:COFFset:
ALARm: SEConds? FREQuency Returns
FREQuency SEConds

**:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:
:FREQuency?**

Description

This query returns the frequency of the input signal in bps.

At *RST, this value is device dependent.

Syntax

:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:
FREQuency? <wsp>FREQuency|FOFFset|
MNOFFset|MPOFFset

:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset :FREQUency?

Parameter(s)

Frequency Type:

The program data syntax for the parameter is defined as a <Character Program Data> element.

The allowed <Character Program Data> elements for this parameter are:

FREQUency|FOFFset|MNOFFset|MPOFFset.

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.

**:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset
:FREQUency?**

Response Syntax <Frequency>

Response(s)

Frequency:

The response data syntax for <Frequency> is defined as a <STRING 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)

SENSe:DATA:TELEcom:ETHernet:COFFset:
FREQUency? FOFFset Returns Frequency

**:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:
:CONFiguration:FOANalysis:ENABLE**

Description	This command sets the enable and disable status of Frequency Offset Analysis.
Syntax	:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset: CONFiguration:FOANalysis:ENABLE <wsp> <Status>
Parameter(s)	Status: The program data syntax for the parameter is defined as a <Boolean Program Data> element. Sets the status of Frequency Offset Analysis.
Example(s)	SENSe:DATA:TELEcom:BERT:COFFset: CONFiguration:FOANalysis:ENABLE 1

:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset: :CONFIguration:FOANalysis:ENABLE?

Description	This query returns the status of Frequency Offset Analysis.
Syntax	:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset: CONFIguration:FOANalysis:ENABLE?
Parameter(s)	None
Response Syntax	<Status>
Response(s)	Status: The response data syntax for <Status> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of Frequency Offset Analysis.
Example(s)	SENSe:DATA:TELEcom:BERT:COFFset: CONFIguration:FOANalysis:ENABLE? Returns 1

**:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset
:CONFIguration:EFRequency?**

Description	<p>This command sets the Expected Frequency (bps) measurements.</p> <p>At *RST, for normal mode, the frequency is set to the configured TX Rate (Refer to TX Rate) and the Frequency Offset Analysis is enabled.</p>
Syntax	:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset: CONFIguration:EFRequency?
Parameter(s)	None
Response Syntax	<Frequency>

**:SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset
:CONFIguration:EFRequency?**

Response(s)	Frequency: The response data syntax for <Frequency> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the Expected Frequency (bps) measurements.
Example(s)	SENSe:DATA:TELEcom:BERT:COFFset:CONFIguration:EFRequency?
See Also	SENSe[1..n]:DATA:TELEcom:ETHernet:COFFset:FREQuency

Multi Channel SDT

**:SENSe[1..n]:DATA:TELEcom:SDT:MONItor:
CONFIg:THREShold:ENABle**

Description

This command enables and enter the SDT threshold value that will be used to declare the pass/fail verdict: 0.001 to 299999.94 ms.

At *RST the value is set to OFF.

Syntax

:SENSe[1..n]:DATA:TELEcom:SDT:MONItor:
CONFIg:THREShold:ENABle <wsp>ON|OFF

**:SENSe[1..n]:DATA:TELEcom:SDT:MONitor:
CONFig:THREshold:ENABle**

Parameter(s)	<p>Staus:</p> <p>The program data syntax for the parameter is defined as a <Boolean Program Data> element.</p> <p>The allowed <Boolean Program Data> elements for this parameter are: ON OFF.</p> <p>Enables and enter the SDT threshold value that will be used to declare the pass/fail verdict.</p> <p>ON, the status of the SDT threshold is set as ON.</p> <p>OFF, the status of the SDT threshold is set as OFF.</p>
Example(s)	<p>SENSe:DATA:TELEcom:SDT:MONitor:CONFig:THREshold:ENABle 1</p>
See Also	<p>SENSe:DATA:TELEcom:SDT:MONitor:CONFig:THREshold 25.0</p> <p>SENSe:DATA:TELEcom:SDT:MONitor:CONFig:THREshold? Returns 25.0</p>

**:SENSe[1..n]:DATA:TELEcom:SDT:MONItor:
CONFIg:THREShold:ENABle?**

Description	The query returns the status of SDT threshold value. At *RST the value is set to OFF.
Syntax	:SENSe[1..n]:DATA:TELEcom:SDT:MONItor: CONFIg:THREShold:ENABle?
Parameter(s)	None
Response Syntax	<Status>

**:SENSe[1..n]:DATA:TELEcom:SDT:MONitor:
CONFig:THREShold:ENABle?**

Response(s)	Status: The response data syntax for <Status> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the status of SDT threshold value.
Example(s)	SENSe:DATA:TELEcom:SDT:MONitor:CONFig:THREShold:ENABle? Returns 1
See Also	SENSe:DATA:TELEcom:SDT:MONitor:CONFig:THREShold 25.0 SENSe:DATA:TELEcom:SDT:MONitor:CONFig:THREShold? Returns 25.0

**:SENSe[1..n]:DATA:TELEcom:SDT:MONItor:
CONFIg:THREShold**

Description	<p>This command enables and enter the SDT threshold value that will be used to declare the pass/fail verdict: 0.001 to 299999.94 ms.</p> <p>At *RST the value is set to 50 ms.</p>
Syntax	<pre>:SENSe[1..n]:DATA:TELEcom:SDT:MONItor: CONFIg:THREShold<wsp><Seconds></pre>
Parameter(s)	<p>Seconds:</p> <p>The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.</p> <p>Configure the SDT threshold value that will be used to declare the pass/fail verdict.</p>
Example(s)	<pre>SENSe:DATA:TELEcom:SDT:MONItor:CONFIg: THREShold 25.0</pre>
See Also	<pre>SENSe:DATA:TELEcom:SDT:MONItor:CONFIg:TH REShold:ENABle 1 SENSe:DATA:TELEcom:SDT:MONItor:CONFIg:TH REShold:ENABle? Returns 1</pre>

**:SENSE[1..n]:DATA:TELEcom:SDT:MONItor:
CONFIg:THREShold?**

Description	The query returns the entered SDT threshold value. At *RST the value is set to 50 ms.
Syntax	:SENSE[1..n]:DATA:TELEcom:SDT:MONItor: CONFIg:THREShold?
Parameter(s)	None
Response Syntax	<Seconds>
Response(s)	Seconds: The response data syntax for <Seconds> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the entered SDT threshold value.
Example(s)	SENSE:DATA:TELEcom:SDT:MONItor:CONFIg:THREShold? Returns 25.0
See Also	SENSE:DATA:TELEcom:SDT:MONItor:CONFIg:THREShold:ENABle 1 SENSE:DATA:TELEcom:SDT:MONItor:CONFIg:THREShold:ENABle? Returns 1

**:SOURce[1..n]:DATA:TELEcom:SDT:CHANnel:
SElect**

Description	This command selects the Channel for SDT measurement. At *RST all the channels are selected.
Syntax	:SOURce[1..n]:DATA:TELEcom:SDT:CHANnel: SElect<wsp><Channel number>, ON OFF

**:SOURce[1..n]:DATA:TELEcom:SDT:CHANnel:
SELEct**

Parameter(s)	<p>Channel number: The program data syntax for the first parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the Channel for SDT measurement.</p> <p>Status: The program data syntax for the second parameter is defined as a <Boolean Program Data> element. The allowed <Boolean Program Data> elements for this parameter are: ON OFF. Selects the status of the Channel for SDT measurement. ON, the status of the channel selection is ON. OFF, the status of the channel selection is OFF.</p>
Example(s)	<p>SOURce:DATA:TELEcom:SDT:CHANnel:SELEct 1,ON</p>
See Also	<p>SOURce:DATA:TELEcom:SDT:CHANnel:SELEct? 1 Returns 1 SOURce:DATA:TELEcom:SDT:CHANnel:ENABLE:STATus? Returns 1</p>

**:SOURce[1..n]:DATA:TELEcom:SDT:CHANnel:
SElect?**

Description	<p>This query returns the selected Channel for SDT measurement.</p> <p>At *RST all the channels are selected.</p>
Syntax	<p>:SOURce[1..n]:DATA:TELEcom:SDT:CHANnel: SElect<wsp><Channel number></p>
Parameter(s)	<p>Channel number: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Selects the Channel for SDT measurement.</p>
Response Syntax	<p><Status></p>

:SOURCE[1..n]:DATA:TELEcom:SDT:CHANnel: SElect?

Response(s)

Status:

The response data syntax for <Status> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the status of the Channel for SDT measurement.

ON, the status of the channel is selected as ON.

OFF, the status of the channel is selected as OFF.

Example(s)

SOURCE:DATA:TELEcom:SDT:CHANnel:SElect? 1
Returns 1

See Also

SOURCE:DATA:TELEcom:SDT:CHANnel:SElect
1,ON

SOURCE:DATA:TELEcom:SDT:CHANnel:ENABLE:S
TATUS? Returns 1

**:SOURce[1..n]:DATA:TELEcom:SDT:CHANnel:
ENABLE:STATus?**

Description	<p>This query returns the status of the Channel for SDT measurement.</p> <p>This command is not associated with any *RST value.</p>
Syntax	:SOURce[1..n]:DATA:TELEcom:SDT:CHANnel: ENABLE:STATus?
Parameter(s)	None
Response Syntax	<Channels>
Response(s)	<p>Channels:</p> <p>The response data syntax for <Channels> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the status of the channels.</p>
Example(s)	SOURce:DATA:TELEcom:SDT:CHANnel:ENABLE:S TATus? Returns 1
See Also	SOURce:DATA:TELEcom:SDT:CHANnel:SELEct? 1 Returns 1 SOURce:DATA:TELEcom:SDT:CHANnel:ENABLE:S TATus? Returns 1

:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:LIST?

Description	<p>This query returns the list of the number of channels.</p> <p>This query is not associated with any *RST value.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:LIST? <wsp><Channel number></p>
Parameter(s)	<p>Channel number: The program data syntax for the parameter is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Displays the list of the number of channels.</p>
Response Syntax	<p>None</p>
Response(s)	<p>None</p>
Example(s)	<p>FETCh:DATA:TELEcom:SDT:RESUlt:LIST? 1</p>
See Also	<p>FETCh:DATA:TELEcom:SDT:RESUlt:LIST? 1</p>

**:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:
CATHreshold?**

Description	<p>This query returns the number of channels that experience a disruption time above the defined threshold since the beginning of the SDT test.</p> <p>At *RST the value is set to 0.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: CATHreshold?
Parameter(s)	None
Response Syntax	<Channel >
Response(s)	<p>Channel:</p> <p>The response data syntax for <Channel> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of channels that experience a disruption time above the defined threshold since the beginning of the SDT test.</p>
Example(s)	FETCh:DATA:TELEcom:SDT:RESUlt: CATHreshold? Returns 0
See Also	FETCh:DATA:TELEcom:SDT:RESUlt:CMONitored? Returns 0

**:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:
CMONitored?**

Description	<p>This query returns the number of channels that are monitored.</p> <p>At *RST the value is set to 0.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: CMONitored?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Channel></p>
Response(s)	<p>Channel:</p> <p>The response data syntax for <Channel> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the number of channels that are monitored.</p>
Example(s)	<p>FETCh:DATA:TELEcom:SDT:RESUlt:CMONitored? Returns 0</p>
See Also	<p>FETCh:DATA:TELEcom:SDT:RESUlt:CATHreshold ? Returns 0</p>

**:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:
LODisruption:DURection?**

Description	<p>This query returns the duration of the longest measured disruption time.</p> <p>At *RST the value is set to --.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: LODisruption:DURection?
Parameter(s)	
Response Syntax	<Time>
Response(s)	<p>Time:</p> <p>The program data syntax for the parameter is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the duration of the longest measured disruption time.</p>
Example(s)	FETCh:DATA:TELEcom:SDT:RESUlt:LODiSruption :DURection? Returns "0"
See Also	FETCh:DATA:TELEcom:SDT:RESUlt:LODiSruption :CHANnel? Returns 0

:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: LODisruption:CHANnel?

Description	<p>This query returns the channel number on which the longest disruption time happened.</p> <p>At *RST the value is set to --.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: LODisruption:CHANnel?</code>
Parameter(s)	None
Response Syntax	<Channel>
Response(s)	<p>Channel:</p> <p>The response data syntax for <Channel> is defined as a <NR1 NUMERIC RESPONSE DATA> element.</p> <p>Returns the channel number on which the longest disruption time happened.</p>
Example(s)	<code>FETCh:DATA:TELEcom:SDT:RESUlt:LODiSruption :CHANnel?</code> Returns 0
See Also	<code>FETCh:DATA:TELEcom:SDT:RESUlt:LODiSruption :DURation?</code> Returns "0"

**:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:
LODisruption:TIMestamp?**

Description	<p>This query returns the time when the longest disruption time happened.</p> <p>At *RST the value is set to --:--:--.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:LODisruption:TIMestamp?
Parameter(s)	None
Response Syntax	<Time>
Response(s)	<p>Time:</p> <p>The response data syntax for <Time> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the time when the longest disruption time happened.</p>
Example(s)	FETCh:DATA:TELEcom:SDT:RESUlt:LODisruption:TIMestamp? Returns Time
See Also	FETCh:DATA:TELEcom:SDT:RESUlt:LODisruption:DURation? Returns "0"

**:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:
VTHReshold?**

Description	<p>This query returns the selected global SDT threshold value when enabled.</p> <p>At *RST the value is set to 0.00083333.</p>
Syntax	<code>:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: VTHReshold?</code>
Parameter(s)	None
Response Syntax	<code><Threshold></code>
Response(s)	<p>Threshold:</p> <p>The response data syntax for <code><Threshold></code> is defined as a <code><NR2 NUMERIC RESPONSE DATA></code> element.</p> <p>Returns the selected global SDT threshold value when enabled.</p>
Example(s)	<code>FETCh:DATA:TELEcom:SDT:RESUlt:VTHReshold?</code> Returns 50.000000
See Also	<code>SENSe:DATA:TELEcom:SDT:MONItor:CONFIg:TH REshold</code>

**:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:
CDISruption?**

Description	<p>This query returns the number of channels that experienced disruptions.</p> <p>At *RST the value is set to 0.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: CDISruption?
Parameter(s)	None
Response Syntax	<Number of channels>
Response(s)	<p>Number of channels: The response data syntax for <Number of channels> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of channels that experienced disruptions.</p>
Example(s)	FETCh:DATA:TELEcom:SDT:RESUlt:CDISruption? Returns 0
See Also	FETCh:DATA:TELEcom:SDT:RESUlt:LADisruption: CHANnel?

**:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:
LADisruption:CHANnel?**

Description	<p>This query returns the channel number that experienced the last disruption.</p> <p>At *RST the value is set to --.</p>
Syntax	<p>:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: LADisruption:CHANnel?</p>
Parameter(s)	<p>None</p>
Response Syntax	<p><Channel></p>
Response(s)	<p>Channel: The response data syntax for <Channel> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the channel number that experienced the last disruption.</p>
Example(s)	<p>FETCh:DATA:TELEcom:SDT:RESUlt:LADisruption:CHANnel? Returns 0</p>
See Also	<p>FETCh:DATA:TELEcom:SDT:RESUlt:LADisruption:TIMestamp?</p>

**:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt:
LADisruption:TIMestamp?**

Description	<p>This query returns the date/time. that experienced the last disruption.</p> <p>At *RST the value is set to --:--:--.</p>
Syntax	:FETCh[1..n]:DATA:TELEcom:SDT:RESUlt: LADisruption:TIMestamp?
Parameter(s)	None
Response Syntax	<Time>
Response(s)	<p>Time:</p> <p>The response data syntax for <Time> is defined as a <STRING RESPONSE DATA> element.</p> <p>Returns the date/time that experienced the last disruption.</p>
Example(s)	FETCh:DATA:TELEcom:SDT:RESUlt:LADisruption: TIMestamp? Returns Time
See Also	FETCh:DATA:TELEcom:SDT:RESUlt:LADisruption: CHANnel?

SONET/SDH SCPI Command Reference

Multi Channel SDT

www.EXFO.com · info@exfo.com

CORPORATE HEADQUARTERS	400 Godin Avenue	Quebec (Quebec) G1M 2K2 CANADA Tel.: 1 418 683-0211 · Fax: 1 418 683-2170
EXFO AMERICA	3701 Plano Parkway, Suite 160	Plano TX, 75075 USA Tel.: 1 972 907-1505 · Fax: 1 972 836-0164
EXFO EUROPE	Omega Enterprise Park, Electron Way	Chandlers Ford, Hampshire S053 4SE ENGLAND Tel.: +44 2380 246810 · Fax: +44 2380 246801
EXFO ASIA-PACIFIC	151 Chin Swee Road #03-29, Manhattan House	SINGAPORE 169876 Tel.: +65 6333 8241 · Fax: +65 6333 8242
EXFO CHINA	No. 88 Fuhua First Road, Central Tower, Room 801, Futian District	Shenzhen 518048 P. R. CHINA Tel.: +86 (755) 8203 2300 · Fax: +86 (755) 8203 2306
	Beijing New Century Hotel Office Tower, Room 1754-1755, No. 6 Southern Capital Gym Road	Beijing 100044 P. R. CHINA Tel.: +86 (10) 6849 2738 · Fax: +86 (10) 6849 2662
EXFO SERVICE ASSURANCE	285 Mill Road	Chelmsford MA, 01824 USA Tel.: 1 978 367-5600 · Fax: 1 978 367-5700
TOLL-FREE	(USA and Canada)	1 800 663-3936

© 2014 EXFO Inc. All rights reserved.
Printed in Canada (2011 -06)



EXFO
EXPERTISE REACHING OUT